Zdzisław Pawlak

SYSTEMY INFORMACYJNE
Podstawy teoretyczne

WYDAWNICTWA NAUKOWO-TECHNICZNE • WARSZAWA 1983

SYSTEMY INFORMACYJNE
Podstawy teoretyczne

Rodzicom

Seria INFORMATYKA

Wykaz książek z serii zamieszczono na końcu książki

Opiniodawcy: WIKTOR MAREK, ADAM WYSOCKI
Redaktor: HALINA TEMPCZYK
Redaktor techniczny: ELŻBIETA GONTARZ
Okładkę i strony tytułowe projektował:
KRZYSZTOF RACINOWSKI

681.3

W książce wprowadzono formalne pojęcie systemu informacyjnego oraz związanego z nim języka. Takie formalne ujęcie służy do badania własności systemów wyszukiwania informacji oraz do precyzowania niektórych problemów sztucznej inteligencji. Książka jest przeznaczona dla projektantów systemów informacyjnych, wykładowców i studentów kierunków informatycznych.

ISBN 83-204-0520-3

WNT Warszawa 1983. Wydanie I. Nakład 4800+200 egz. Ark. wyd. 12,6. Ark. druk. 11,75 (15,63/A). Format B5. Papier satynowany kl. III 70 g. Oddano do składania 12 VI 1983 r. Podpisano do druku w grudniu 1983 r. Druk ukończono w grudniu 1983 r. Symbol IPT/79877

Druk: Bielskie Zakłady Graficzne, Bielsko-Biała, zam. 1525/83 M-29

SPIS TREŚCI

PRZEDMOWA 9

1. SYSTEMY INFORMACYJNE 13
   1.1. Wprowadzenie 13
   1.2. Definicja systemu informacyjnego 14
   1.3. Własności systemów informacyjnych 19
   1.4. Zależność atrybutów 26
   1.5. Systemy zredukowane 33
   1.6. Podsystemy informacyjne 36
   1.7. Łączenie systemów informacyjnych 40

2. JĘZYKI SYSTEMÓW INFORMACYJNYCH 48
   2.1. Wprowadzenie 48
   2.2. Składnia języka 51
   2.3. Semantyka języka 53
   2.4. Reguły przekształcania termów 58
   2.5. Postać normalna termów 60
   2.6. Dokładność i efektywność języka 64
   2.7. Instrukcje 67
   2.8. Instrukcje liczbowe 68
   2.9. Instrukcje numeryczne 69
   2.10. Instrukcje relacyjne 73
   2.11. Instrukcje logiczne 75
   2.12. Instrukcje porządkujące 77

3. SYSTEMY WIELOSTOPNIOWE I HIERARCHICZNE 79
   3.1. Wprowadzenie 79
   3.2. Systemy dwustopniowe 79
   3.3. Systemy wielostopniowe 83
   3.4. Systemy hierarchiczne 84
   3.5. Uwagi końcowe 86

4. SYSTEMY ROZPROSZONE 87
   4.1. Wprowadzenie 87
   4.2. System rozproszony i jego języki 88
   4.3. System z rozproszonymi atrybutami 89
   4.4. System z rozproszonymi obiektami 93
   4.5. Użytkownik centralny a użytkownik lokalny 94
   4.6. Pamięć centralna i pamięć lokalna 96
   4.7. Uwagi końcowe 98

5. SYSTEMY INFORMACYJNE WIELOWARTOŚCIOWE 100

5.1. Wprowadzenie 100
5.2. Przykład systemu wielowartościowego 100
5.3. Definicja systemu informacyjnego wielowartościowego 101
5.4. Składnia języka systemów wielowartościowych 105
5.5. Semantyka języka systemów wielowartościowych 105
5.6. Aksjomaty języka wielowartościowego 108

6. PRZYBLIŻONE SYSTEMY INFORMACYJNE 111

6.1. Wprowadzenie 111
6.2. Definicje przybliżonego systemu informacyjnego 112
6.3. Przykład systemu przybliżonego 112
6.4. Składnia języka przybliżonego systemu informacyjnego 114
6.5. Semantyka języka przybliżonego systemu informacyjnego 115
6.6. Obliczanie wartości termów w języku przybliżonym 117
6.7. Obliczanie przybliżone wartości termów języka dokładnego 122

7. STOCHASTYCZNE SYSTEMY INFORMACYJNE 125

7.1. Wprowadzenie 125
7.2. Stochastyczny system informacyjny 126
7.3. Język stochastycznych systemów informacyjnych 128
7.4. Podsumowanie 132

8. PRZYBLIŻONA KLASYFIKACJA OBIEKTÓW 134

8.1. Wprowadzenie 134
8.2. Zbiory przybliżone 135
8.3. Własności przybliżeń 140
8.4. Przybliżony opis podzbiorów 145
8.5. Próbka zbioru 148
8.6. Przykład zastosowania 149
8.7. Klasyfikacja wielowartościowa 151
8.8. Dokładność przybliżonej klasyfikacji 153
8.9. Uwagi końcowe 154

9. NIEPEŁNA KLASYFIKACJA OBIEKTÓW 155

9.1. Uwagi wstępne 155
9.2. Niepełna klasyfikacja dwuwartościowa 155
9.3. Przybliżona równość zbiorów 157
9.4. Przypadek wielu ekspertów 160

10. O PROCESIE UCZENIA SIĘ 163

10.1. Uwagi wstępne 163
10.2. Proces uczenia 163
10.3. Dokładność uczenia się — oszacowanie od dołu 165
10.4. Uwagi końcowe 167

ZAKOŃCZENIE 168

Dodatek A. POJĘCIA MATEMATYCZNE UŻYWANE W PRACY 170

A1. Notacja logiczna 170
A2. Zbiory 170
A3. Relacje 171
A4. Relacje równoważności 172
A5. Relacje porządku 172
A6. Funkcje 172

Dodatek B. UWAGI O REALIZACJI SYSTEMÓW INFORMACYJNYCH METODĄ SKŁADOWYCH ATOMOWYCH 174

B1. Uwagi wstępne 174
B2. Zbiory elementarne 174
B3. Wyszukiwanie zbiorów elementarnych. 176
B4. Postać normalna 177
B5. Wielodostęp 177
B6. Systemy rozproszone 177
B7. Konsekwencje sprzętowe. 178

LITERATURA 179

SKOROWIDZ 184

PRZEDMOWA

W niniejszej książce są rozważane dwa zasadnicze zagadnienia. Pierwsze z nich to, jak znaleźć obiekty, gdy opis ich własności jest dany w pewnym języku formalnym. Drugie jest w pewnym sensie zagadnieniem odwrotnym: dany jest zbiór obiektów, a należy znaleźć opis charakterystyczny tego zbioru w zadanym języku formalnym. Oba te problemy mają zasadnicze znaczenie we współczesnych zastosowaniach maszyn liczących.

Pierwszy z nich dotyczy systemów wyszukiwania informacji, drugi zaś jest związany z ważnym działem zastosowań maszyn liczących, a mianowicie z tzw. sztuczną inteligencją i wiąże się przede wszystkim z problemami automatyzacji rozumowania indukcyjnego.

Punktem wyjścia do obu tych problemów jest pojęcie systemu informacyjnego oraz jego języka, dlatego wyszukiwanie informacji oraz rozumowanie indukcyjne, zagadnienia na pierwszy rzut oka bardzo odległe, znalazły się w tym tekście obok siebie.

Celem niniejszego opracowania nie jest przedstawienie stanu tych dwu dziedzin, lecz raczej sprecyzowanie niektórych podstawowych pojęć dotyczących wyszukiwania informacji oraz rozumowania indukcyjnego.

Istnieje bardzo bogata literatura dotycząca podstaw teoretycznych wyszukiwania informacji oraz rozumowania indukcyjnego, jednakże w zasadzie nie będziemy się do niej odwoływać, gdyż przedstawione tu podejście różni się dość istotnie od dotychczasowego traktowania tych zagadnień. Punktem wyjścia do nowych rozważań jest, jak to wspomniano poprzednio, pojęcie systemu informacyjnego oraz związanego z nim języka formalnego; pojęcia te były sformułowane przez autora [89] oraz zmodyfikowane w pracy Marka i Pawlaka [63], a następnie były rozwijane w kilku ośrodkach w kraju, stąd niemal cała literatura dotycząca tego tematu jest głównie pióra autorów polskich.

Podane tu pojęcie systemu informacyjnego jest dość bliskie modelom rozważanym przez Maedę [56], Saltona [103], Wonga i Chianga [115], Babada [3], Cherniavskiego i Schneidera [8], a przede wszystkim Codda [9], jednakże nie pokrywa się ono całkowicie ze wspomnianymi modelami. Różnic tych jednak nie będziemy tu dyskutowali. Zainteresowany Czytelnik z łatwością może wyrobić sobie pogląd — na podstawie cytowanej literatury — na temat podobieństw i różnic między proponowaną definicją systemu informacyjnego a innymi modelami systemów informacyjnych.

Wprowadzone pojęcie systemu informacyjnego zostało użyte nie tylko do zbadania pewnych własności systemów wyszukiwania informacji, lecz także do sprecyzowania niektórych problemów związanych z automatyzacją rozumowania indukcyjnego. Ta część pracy została zainspirowana wynikami Michalskiego [79], i wykazuje największe podobieństwo do jego podejścia do tego zagadnienia. Główna różnica między proponowanym podejściem do problematyki rozumowania indukcyjnego a podejściem Michalskiego i innych polega na przyjęciu jako punktu wyjścia zdefiniowanego pojęcia systemu informacyjnego. Pozwoliło to między innymi na proste określenie dokładności rozumowania (w naszej terminologii — dokładności uczenia), co, jak się wydaje, nie byłoby tak proste w innych modelach.

Książka ta jest przeznaczona przede wszystkim dla studentów i wykładowców, jako podręcznik pomocniczy do przedmiotów „systemy informacyjne" oraz „sztuczna inteligencja". Pewne partie mogą również zainteresować projektantów systemów informacyjnych oraz systemów klasyfikacyjnych, rozpoznających czy też uczących się. Chociaż w tej książce nie mówi się bezpośrednio o praktycznych zastosowaniach podanych idei, wiele z nich może znaleźć, a niektóre już znalazły takie zastosowanie.

Z podanych tu własności systemu informacyjnego wynika nowa metoda wyszukiwania informacji. Jest ona dość zbliżona do metody zaproponowanej przez Wonga i Chianga [115], chociaż zachodzą między nimi również istotne różnice. Występują też w niektórych koncepcjach daleko idące podobieństwa zaproponowanego ujęcia rozwiązań w modelu relacyjnym Codda, podejścia te jednak w istocie różnią się zasadniczo.

Proponowana metoda wyszukiwania informacji została nazwana roboczo *metodą składowych atomowych.* Została ona zrealizowana i zbadana od strony użytkowej, najpierw przez Margańskiego [71], a następnie niezależnie przez Błaszczyka [5] i Telukową [107]. Zrealizowane na jej podstawie systemy pracują od dłuższego czasu użytkowo i wykazują wiele zalet eksploatacyjnych. Kilka dalszych realizacji eksperymentalnych jest w toku, między innymi w Instytucie Matematyki Politechniki Warszawskiej, pod kierunkiem mgra K. Grzelaka.

Metoda ta została również z powodzeniem zastosowana przez A. T. Francisa do statystycznej bazy danych — na podstawie koncepcji podanych w pracach [4] i [7][1].

Także z rozważań nad rozumowaniem indukcyjnym wynikają nowe metody realizacji systemów rozpoznających, klasyfikujących i innych. Jedna z takich metod została zrealizowana [11] w celach eksperymentalnych, nie została ona jednak do tej pory w pełni zbadana.

Od strony teoretycznej podane tu pojęcia zostały rozszerzone w różnych kierunkach (patrz np. prace [22], [23], [48], [49], [108], [85], [86], [82], [111], [3], [6], jednakże jak dotąd nie zbadano możliwości ich praktycznego wykorzystania.

---

[1] Patrz A. T. Francis: Application of the atomic constituents method to statistical data base, praca doktorska, Instytut Podstaw Informatyki PAN, 1982.

Lektura książki nie wymaga w zasadzie żadnego przygotowania specjalistycznego, jednakże osoby, które mają wiedzę praktyczną z zakresu budowy maszyn liczących i ich zastosowań, a w szczególności ci, którzy zrealizowali samodzielnie jakiś system, na pewno znajdą tu znacznie więcej materiału do przemyśleń niż Czytelnicy bez takiego doświadczenia, którzy będą w stanie zrozumieć jedynie warstwę „zewnętrzną" podanych rozważań.

Do zrozumienia książki jest potrzebna znajomość matematyki nie przekraczająca zasobu wiedzy absolwenta szkoły średniej. Czytelnikowi nie znającemu używanego tu formalizmu polecam książki Rasiowej [95], Kuratowskiego [36] oraz Marka i Onyszkiewicza [59], w których można znaleźć szczegółowe wyjaśnienie użytych pojęć i notacji.

Książka składa się z dziesięciu rozdziałów i dwu dodatków. Rozdziały od 1 do 7 dotyczą systemów wyszukiwania informacji, rozdziały 8÷10 rozumowania indukcyjnego. Do zrozumienia rozdziałów 8÷10 jest potrzebna znajomość rozdziałów 1, 2, w których są wyjaśnione pojęcia systemu informacyjnego oraz jego języka — stanowiące punkt wyjścia do rozważań nad systemami wyszukiwania informacji oraz do rozumowań indukcyjnych.

W dodatku A podano w skrócie wyjaśnienia najważniejszych pojęć matematycznych użytych w pracy, dodatek B zawiera pewne ogólne informacje związane z realizacją podanych tu koncepcji.

Pragnę wyrazić podziękowanie za wiele cennych uwag pierwszym Czytelnikom maszynopisu: Małgorzacie Pawlak, dr Alicji Wakulicz-Deja, prof. Mirosławowi Dąbrowskiemu oraz prof. Konradowi Fiałkowskiemu. Szczególną wdzięczność jestem winien doc. Ewie Orłowskiej oraz prof. Wiktorowi Markowi, których uwagi pozwoliły mi uniknąć wielu istotnych niedociągnięć i błędów.

Zdzisław Pawlak

Warszawa, październik 1981

1. SYSTEMY INFORMACYJNE

1.1. WPROWADZENIE

Zanim rozpoczniemy ściślejsze rozważania na temat systemów informacyjnych, rozpatrzymy najpierw kilka prostych przykładów takich systemów, na których postaramy się pokazać główne problemy występujące w omawianej dziedzinie.

Zaczniemy od książki telefonicznej. Jeśli chcemy znaleźć numer telefonu np. Jana Kowalskiego, zamieszkałego w Warszawie przy ul. Filtrowej, wystarczy zajrzeć w odpowiednie miejsce książki telefonicznej; o ile oczywiście numer ten znajduje się w książce telefonicznej.

Jeśli numeru telefonu interesującej nas osoby w spisie telefonów nie znaleźliśmy, nie wynika z tego, że osoba ta telefonu nie ma. Mogła ona założyć telefon niedawno i dlatego numer nie znalazł się jeszcze w aktualnym spisie, może też to oznaczać, że szukany numer telefonu jest zastrzeżony i nie został zamieszczony w powszechnie dostępnym spisie abonentów.

Pojawiły się tu od razu dwa ważne problemy związane z systemami informacyjnymi: aktualności informacji oraz dostępności informacji.

System informacyjny, aby był użyteczny, musi zawierać aktualne informacje. Do osiągnięcia tego celu, niezależnie od środków administracyjnych, są konieczne odpowiednie środki techniczne — szczególnie w przypadku skomputeryzowanych systemów informacyjnych.

Nie wszystkie informacje zawarte w systemie są jednakowo dostępne. Pewne informacje mogą być dostępne tylko dla określonych osób. Jest to problem ochrony informacji w systemie przed nieodpowiednim ich wykorzystaniem.

W rozpatrywanym przykładzie systemu informacji telefonicznej sprawa aktualności informacji oraz dostępności informacji nie ma może bardzo dużej wagi. Problem aktualności informacji odgrywa natomiast znacznie większą rolę w systemie informacji medycznej zawierającym dane o aktualnym stanie pacjentów znajdujących się np. w szpitalu. Aktualność informacji może mieć tu zasadnicze znaczenie i ten aspekt wysuwa się na pierwszy plan w tego rodzaju systemach.

Problem ograniczenia dostępu do informacji odgrywa zasadniczą rolę w systemach informacji wojskowej, np. na temat uzbrojenia. W takim systemie problem ścisłej kontroli dostępu do informacji jest sprawą najważniejszą.

Wróćmy jeszcze do informacji telefonicznej. Gdybyśmy chcieli na podstawie spisu telefonów uzyskać informację, kto jest posiadaczem telefonu o określonym numerze, należałoby po prostu odszukać w spisie odpowiedni numer i odczytać dane o jego właścicielu (z problemem takim spotyka się np. milicja). Uzyskanie tego rodzaju informacji byłoby znacznie bardziej uciążliwe niż poszukiwanie numeru telefonu osoby, której nazwisko i adres są znane. W pierwszym przypadku należałoby bowiem przeglądać kolejno wszystkie pozycje w spisie telefonów. Oczywiście, można by ułożyć spis abonentów w kolejności ich numerów telefonicznych, lecz to utrudniłoby wyszukiwanie informacji drugiego rodzaju.

Już na przykładzie książki telefonicznej widać zatem, że znalezienie pożądanej informacji może być mniej lub bardziej skomplikowane. Aby uzyskać łatwy dostęp do informacji w systemie, należy z góry znać rodzaj pytań, na które będziemy chcieli znaleźć odpowiedź, i nie tylko zamieścić w systemie potrzebne, właściwe informacje, lecz także odpowiednio ów system „zorganizować".

Podane tu problemy występują w każdym systemie informacyjnym. W różnych systemach mają one jednak różną wagę. W jednych systemach najważniejsza może być aktualność informacji, w innych efektywność wyszukiwania informacji, w jeszcze innych — ograniczenie dostępu do informacji tylko dla uprawnionych użytkowników systemu.

Reasumując, w każdym systemie informacyjnym spotykamy się z następującymi problemami:

(1) klasą pytań, na które system umożliwia odpowiedzi;
(2) rodzajem informacji zawartych w systemie;
(3) organizacją systemu;
(4) aktualnością zawartej w systemie informacji;
(5) efektywnością systemu;
(6) ochroną przed niepowołanym dostępem.

W książce tej nie będziemy zajmować się wszystkimi wymienionymi powyżej problemami, lecz skupimy uwagę w zasadzie na początkowych trzech problemach, ściślej zajmiemy się sprecyzowaniem podstawowych pojęć związanych z systemami informacyjnymi i zbadaniem ich najważniejszych własności.

1.2. DEFINICJA SYSTEMU INFORMACYJNEGO

W tym punkcie podamy formalną definicję systemu informacyjnego. Pierwszą taką definicję podał Salton [101], ma ona charakter bardzo ogólny i określa szeroką klasę systemów informacyjnych. Definicja, którą przyjmiemy w tej książce — mimo że jest szczególnym przypadkiem definicji Saltona — różni się od niej w dość istotny sposób, przede wszystkim z uwagi na położenie dużego nacisku na stronę języka zapytań związanego z systemem informacyjnym. Przyjęta tu definicja systemu informacyjnego została po raz pierwszy sformułowana w pracy Pawlaka [87], a następnie w zmodyfikowanej postaci w pracy Marka i Pawlaka [61]. Ostateczna wersja definicji, używanej w niniejszej książce została po raz pierwszy zamieszczona w pracy [93].

W każdym systemie informacyjnym wyróżniamy zbiór (skończony oczywiście) obiektów, o których system ma zawierać informacje, takich jak np. pracownicy, chorzy, książki, części itd. Zakładamy ponadto, że w każdym konkretnym systemie jest to zbiór elementów jednego rodzaju, a więc bądź to ludzi, bądź książek, bądź też maszyn, tzn. w systemie nie występują jednocześnie obiekty różnego rodzaju, np. ludzie i książki lub książki i maszyny.

Ponieważ każdy system ma zawierać pewne informacje o ustalonych obiektach, musimy w definicji systemu dysponować odpowiednimi środkami formalnymi do sprecyzowania tego faktu. Rodzaj informacji zależy oczywiście od tego, o co będziemy chcieli pytać. Rodzaj pytań określa więc rodzaj informacji o obiektach zawartych w systemie. W naszych rozważaniach zajmiemy się najprostszym i najczęściej spotykanym przypadkiem, systemu, w którym obiekty charakteryzuje się przez ich własności. W tym celu będziemy używali pojęcia *atrybutu* oraz *wartości atrybutu*. Atrybutem może być np. kolor oczu, wzrost, nazwisko, adres zamieszkania, płeć, stan cywilny itd. Kolor oczu może być niebieski, piwny, zielony; wzrost może być dowolną liczbą rzeczywistą z określonego przedziału, np. 170 cm, albo też określeniem: niski, średni, wysoki (co odpowiada pewnemu przedziałowi liczbowemu). Wartością atrybutu „nazwisko" może być np. Kowalski.

Przyjmiemy dalej, że każdy obiekt w naszym systemie jest charakteryzowany przez podanie wartości ustalonego dla danego systemu zbioru atrybutów. Na przykład w systemie ewidencji ludności każdy człowiek jest charakteryzowany przez nazwisko, imię, datę urodzenia, miejsce urodzenia, adres zamieszkania, wzrost, kolor oczu, stan cywilny. Są to po prostu rubryki, które występują w każdym dowodzie osobistym i dla każdej osoby muszą one być odpowiednio wypełnione.

Zwróćmy uwagę, że jak już wspominaliśmy, sprawa doboru atrybutów i ich wartości jest związana z rodzajem pytań, jakie chcemy stawiać systemowi; również rodzaj atrybutów jest związany z rodzajem obiektów występujących w systemie. Atrybut „płeć" może być odniesiony do ludzi lub zwierząt, nie może być natomiast używany w przypadku maszyny lub dokumentu. Natomiast atrybut „waga" może być używany zarówno w odniesieniu do ludzi, jak i maszyn. Rodzaj obiektów wyznacza w pewien sposób rodzaj atrybutów, które mogą być z nim związane, jednakże tą sprawą nie będziemy się zajmować.

W każdym systemie informacyjnym będziemy wyróżniać skończony zbiór obiektów X, i skończony zbiór atrybutów A. Z każdym atrybutem a należącym do A zwiążemy zbiór jego wartości V_a, który będziemy też czasem nazywać *dziedziną* atrybutu a. Przyjmiemy, że dziedzina każdego atrybutu jest co najmniej dwuelementowa, tzn., każdy atrybut może przyjmować co najmniej jedną z dwu możliwych wartości. Oczywiście niektóre atrybuty mogą mieć wspólne wartości, np. dla atrybutów „długość" oraz „szerokość" zbiorem wartości są liczby rzeczywiste.

Dla opisu własności obiektów systemu wprowadzimy dwuargumentową funkcję ϱ, która każdemu obiektowi $x \in X$ i atrybutowi $a \in A$ przyporządkuje wartość v należącą do dziedziny V_a atrybutu a.

Możemy teraz podać formalną definicję systemu informacyjnego [93].

Definicja

Przez *system informacyjny* będziemy rozumieli czwórkę

$$S = \langle X, A, V, \varrho \rangle$$

gdzie: X — skończony zbiór obiektów; A — skończony zbiór atrybutów; $V = \bigcup_{a \in A} V_a$, V_a dziedzina atrybutu a; ϱ — funkcja całkowita, $\varrho \colon X \times A \to V$; przy czym $\varrho(x, a) \in V_a$ dla każdego $x \in X$ oraz $a \in A$. ■

Zauważmy, że przez warunek, aby funkcja ϱ była całkowita (tzn. określona dla wszystkich wartości argumentów x oraz a), żądamy zawsze znajomości wartości każdego atrybutu. A więc niedopuszczalny jest przypadek, kiedy wartość jakiegoś atrybutu nie jest podana, tzn. każda rubryka w dokumencie opisującym obiekt systemu musi być wypełniona. Przypadek, gdy warunek ten nie jest spełniony, omówiono w rozdz. 5.

Wyjaśnimy bliżej pojęcie dziedziny atrybutu.

Definicja

Dziedziną V_a atrybutu a w systemie S jest zbiór V_a określony następująco:

$$V_a = \{v \in V\colon \text{dla których istnieje } x \in X \text{, takie że } \varrho(x, a) = v\}$$ ■

Zauważmy, że przy takim rozumieniu dziedziny atrybutu wartościami atrybutu, np. KOLOR, w danym systemie informacyjnym nie są wszystkie możliwe kolory, lecz tylko te, które w danym systemie występują. Oczywiście w innym systemie informacyjnym ten sam atrybut może mieć inne wartości.

Przykład 1.1

Rozpatrzmy prosty przykład systemu informacyjnego, którego obiektami są ludzie opisywani za pomocą atrybutów: NAZWISKO, IMIĘ, MIEJSCE URODZENIA, ROK URODZENIA, STAN CYWILNY (nazwy atrybutów będziemy w dalszym ciągu zawsze pisać dużymi literami).

Zbiory wartości poszczególnych atrybutów są oczywiste. Funkcję ϱ można określić za pomocą następującej tabelki:

X	NAZWISKO	IMIĘ	MIEJSCE URODZENIA	ROK URODZENIA	STAN CYWILNY
x_1	Kowalski	Jan	Kraków	1958	kawaler
x_2	Nowak	Jerzy	Warszawa	1930	wdowiec
x_3	Lipiński	Gabriel	Łódź	1910	żonaty
x_4	Mostowicz	Euzebiusz	Wilno	1926	rozwiedziony
...	...	...	...	...	...
x_{357}	Baran	Łukasz	Gdańsk	1970	kawaler

Zwróćmy uwagę, że różne atrybuty mogą mieć znacznie różniące się liczby wartości. STAN CYWILNY — kilka wartości, ROK URODZENIA — co najwyżej 150, pozostałe zaś wiele tysięcy. ■

Przykład 1.2

System ewidencji samochodów może zawierać następujące atrybuty: MARKA SAMOCHODU, RODZAJ SAMOCHODU, ROK PRODUKCJI, POJEMNOŚĆ SILNIKA, WŁAŚCICIEL, KOLOR, NUMER SILNIKA, NUMER NADWOZIA. Określenie funkcji ϱ polega po prostu na podaniu wartości atrybutów dla wszystkich interesujących nas samochodów. Podobnie jak poprzednio, atrybuty mogą mieć różną liczbę wartości. Niektóre atrybuty mają wspólne wartości. Na przykład ROK PRODUKCJI, MOC SILNIKA, NUMER NADWOZIA i NUMER SILNIKA przyjmują jako wartości liczby naturalne. ■

Parę (a, v), gdzie a jest atrybutem, $v \in V_a$ zaś jest wartością atrybutu a należącą do jego dziedziny, będziemy nazywali *deskryptorem*. Deskryptorami są na przykład następujące pary (NAZWISKO, Kowalski), (ROK URODZENIA, 1980), (KOLOR OCZU, niebieski) itd.

Wprowadzimy jeszcze kilka pojęć potrzebnych nam w dalszych rozważaniach. Dla każdego $x \in X$ przez ϱ_x będziemy rozumieć funkcję o argumentach w A oraz wartościach w V, taką że

$$\varrho_x(a) = \varrho(x, a)$$

Funkcję ϱ_x będziemy nazywali *informacją o obiekcie* x w systemie S. Zamiast „informacja o obiekcie" będziemy też czasem mówić *dane o obiekcie.*

Informacja o obiekcie w danym systemie jest to po prostu zbiór wartości wszystkich atrybutów obiektu w danym systemie. Na przykład w systemie rozpatrywanym w przykładzie 1.1 informacja o obiekcie x_2 ma postać:

$\varrho_{x_2} =$

NAZWISKO	IMIĘ	MIEJSCE PRACY	ROK URODZENIA	STAN CYWILNY
Nowak	Jerzy	Warszawa	1930	wdowiec

Mówiąc niezbyt dokładnie, każdy wiersz tablicy funkcji ϱ jest informacją o obiekcie znajdującym się w tym wierszu. Mówiąc inaczej, informacja o obiekcie jest zbiorem deskryptorów. Poprzednia informacja ϱ_{x_2} jest zbiorem następujących deskryptorów:

(NAZWISKO, Nowak), (IMIĘ, Jerzy), (MIEJSCE PRACY, Warszawa), (ROK URODZENIA, 1930), (STAN CYWILNY, wdowiec)

Zbiór deskryptorów wyznaczony przez informację o obiekcie x będziemy nazywać *deskrypcją* albo *opisem obiektu* x w systemie S.

Różnica między informacją o obiekcie a jego opisem ma charakter jedynie formalny. Informacja o obiekcie jest pewną funkcją, natomiast opis obiektu jest pewnym tworem językowym wyznaczonym jednoznacznie przez informację. Dodajmy jeszcze, że kolejność deskryptorów w opisie jest nieistotna. Warto również zauważyć, że gdy mówimy o informacji, zbiory A i V traktujemy jako pewne twory abstrakcyjne, gdy zaś posługujemy się pojęciem opisu elementy zbiorów, A i V traktujemy jako nazwy pewnych pojęć. Na przykład deskryptor (KOLOR OCZU, orzechowe) w języku angielskim będzie miał postać (COLOUR OF EYES, hazel). Gdy mówimy więc o informacji, nie interesuje nas, w jakim języku jest ona wyrażana, natomiast opisy odpowiadające tej samej informacji w różnych językach mogą być różne. W dalszym ciągu nie będziemy tych spraw ściśle rozróżniać, gdyż zawsze będzie wiadomo, kiedy mamy na myśli jedno bądź drugie. Zresztą ścisłe odróżnianie pojęć od ich nazw jest bardzo uciążliwe i w naszych rozważaniach nie jest konieczne.

Na zakończenie tego punktu dodajemy jeszcze, że w myśl przyjętej definicji informacja o obiekcie jest wyczerpująca i wyłączająca. Znaczy to, że wartości każdego atrybutu wyczerpują wszystkie możliwości i tylko jedna wartość atrybutu może być przypisana każdemu obiektowi. Jest rzeczą oczywistą, że pierwszy warunek musi być zawsze spełniony w praktyce, drugi natomiast jest pewnym uproszczeniem, które nie zawsze może być przyjęte przy rozpatrywaniu rzeczywistych systemów informacyjnych. Na przykład wartością atrybutu AUTOR może być nie jedno nazwisko, lecz kilka — gdy jakaś książka jest napisana przez kilku autorów.

W dalszych rozdziałach pokażemy, jak również taki przypadek może być włączony do naszych rozważań.

Przyjmujemy również w naszej definicji systemu informacyjnego, że zbiory obiektów, atrybutów i wartości atrybutów są raz na zawsze ustalone, co w rzeczywistych systemach również nie zachodzi. Zbiór obiektów w rzeczywistym systemie może być zwiększony lub zmniejszony (np. książki w bibliotece bądź pacjenci w szpitalu). Możemy również być zainteresowani usuwaniem lub dołączaniem nowych atrybutów do systemu. Na przykład w rejestrze samochodów może się pojawić atrybut CZYSTOŚĆ SPALIN, który do tej pory nie był uwzględniany. Często również są zmieniane wartości atrybutów. Adres może ulegać zmianie, wzrost, stan cywilny, temperatura, ciśnienie, są też atrybutami, których wartości ulegają zmianom. Nasza definicja takich zmian nie dopuszcza. Mimo tych dość istotnych ograniczeń przyjęta definicja jest użyteczna, pozwala bowiem na sprecyzowanie wielu pojęć dotyczących systemów informacyjnych i zbadanie wielu ich podstawowych własności.

Na zakończenie zauważmy, że w istocie ze względu na skończoną liczbę elementów zbioru obiektów, atrybutów i własności atrybutów, każdy system informacyjny można utożsamiać z tablicą, w której kolumny, poza pierwszą, są oznaczone odpowiednimi atrybutami, w kolumnach są kolejno wpisane wartości atrybutów, w pierwszej kolumnie zaś są podane obiekty systemu. Każdy wiersz tablicy jest informacją o znajdujących się w tym wierszu obiektach. Ze względu na warunek całkowitości funkcji ϱ wszystkie wiersze (kolumny) muszą być odpowiednimi wartościami atrybutów.

W dalszym ciągu system informacyjny możemy więc utożsamiać z tablicą, taką jak w przykładzie 1.1.

1.3. WŁASNOŚCI SYSTEMÓW INFORMACYJNYCH

W tym punkcie podamy kilka podstawowych własności systemów informacyjnych, które pozwolą nam je lepiej zrozumieć.

W poprzednim punkcie wprowadziliśmy pojęcie informacji o obiekcie w systemie. Do sformułowania niektórych własności systemu będzie nam potrzebne pojęcie informacji w systemie, niezależnie od tego, jakiego obiektu systemu informacja ta ma dotyczyć.

Definicja

Informacją w systemie S będziemy nazywać każdą funkcję φ o argumentach w zbiorze atrybutów A oraz o wartościach należących do V, taką że $\varphi(a) \in V_a$. ■

Zbiór wszystkich możliwych informacji w systemie S oznaczymy przez $\mathrm{Inf}(S)$.

Ponieważ w naszej definicji systemu informacyjnego przyjęliśmy, że wszystkie zbiory X, A, V są skończone, więc w każdym systemie może być tylko skończona liczba różnych informacji. Wszystkich możliwych (różnych) informacji w systemie jest oczywiście

$$\prod_{a \in A} \mathrm{card}(V_a)$$

Na przykład, jeżeli w systemie S występują trzy atrybuty a_1, a_2, a_3 oraz atrybut a_1 może przyjmować dwie wartości, atrybut a_2 — trzy wartości, atrybut a_3 — również trzy wartości, to system taki ma $2 \cdot 3 \cdot 3 = 18$ różnych informacji. Informacją w systemie S mogą być na przykład następujące opisy:

$$(a_1, v_1),\ (a_2, v_2),\ (a_3, v_3)$$

$$(a_1, u_1),\ (a_2, u_2),\ (a_3, u_3)$$

$$(a_1, w_1),\ (a_2, w_2),\ (a_3, w_3)$$

gdzie v_1, u_1, w_1 są pewnymi wartościami atrybutu a_1; wartości v_2, u_2, w_2 należą do dziedziny atrybutu a_2, zaś v_3, u_3, w_3 są wartościami atrybutu a_3.

Każda informacja φ wyznacza pewien zbiór obiektów X_φ, takich że

$$X_\varphi = \{x \in X: \varphi_x = \varphi\}$$

tzn. obiektów mających w systemie S jednakową informację (jednakowy opis). Przedmioty o tym samym opisie w systemie S są w tym systemie nierozróżnialne (identyczne). W systemach informacyjnych, np. telefonicznych, chcemy aby każda informacja dotyczyła tylko jednego obiektu — czasem jednak warunek ten nie musi być spełniony. Na przykład w systemie informacji bibliotecznej możemy być zainteresowani otrzymaniem wszystkich książek na określony temat, stąd danemu opisowi może odpowiadać więcej niż jeden obiekt (książka). Może również zaistnieć przypadek, w którym danej informacji nie odpowiada żaden obiekt w systemie. Będziemy mówili wtedy, że dana informacja jest *pusta*. Informacja φ jest pusta, gdy $X_\varphi = \emptyset$. W przeciwnym przypadku informację będziemy nazywali *niepustą*.

System informacyjny, w którym każda informacja jest niepusta, będziemy nazywali *systemem zupełnym* (albo *kompletnym*). Natomiast system, w którym każdej informacji odpowiada co najwyżej jeden obiekt, będziemy nazywali *systemem selektywnym.*

System informacji telefonicznej jest selektywny, natomiast system informacji bibliotecznej (czy patentowej) jest na ogół nieselektywny.

Przykład 1.3

Rozpatrzmy system informacyjny $S=\langle X, A, V, \varrho\rangle$, w którym

$$X=\{x_1, x_2, x_3, x_4\}$$
$$A=\{a, b, c\}$$
$$V_a=\{p_1, p_2\}$$
$$V_b=\{q_1, q_2, q_3\}$$
$$V_c=\{r_1, r_2, r_3\}$$

funkcja ϱ zaś jest określona za pomocą tablicy

X	a	b	c
x_1	p_1	q_2	r_1
x_2	p_1	q_3	r_2
x_3	p_1	q_2	r_1
x_4	p_2	q_1	r_3

Funkcja φ, taka że $\varphi(a)=p_1$, $\varphi(b)=q_2$, $\varphi(c)=r_1$, lub opis (a, p_1), (b, q_2), (c, r_1) jest informacją w systemie S, oraz $X_\varphi=\{x_1, x_3\}$ — co łatwo zauważyć z tabelki, lub też można policzyć, jak to pokazano poniżej

$$X_\varphi=\{x\in X\colon \varrho_x=\varphi\}=$$
$$=\{x\in X\colon \bigwedge_{a\in A}\varrho_x(a)=\varphi(a)\}=$$
$$=\bigcap_{a\in A}\{x\in X\colon \varrho(x,a)=\varphi(a)\}=$$
$$=\{x\in X\colon \varrho(x,a)=p_1\}\cap\{x\in X\colon \varrho(x,b)=q_2\}\cap$$
$$\cap\{x\in X\colon \varrho(x,c)=r_1\}=$$
$$=\{x_1, x_2, x_3\}\cap\{x_1, x_3\}\cap\{x_1, x_3\}=$$
$$=\{x_1, x_3\}$$

System ten nie jest ani selektywny, ani kompletny, gdyż obiekty x_1, x_3 mają jednakową informację w tym systemie oraz istnieją w nim informacje, którym nie odpowiadają żadne obiekty w systemie, np. (a, p_1), (b, q_1), (c, r_1). ∎

Wprowadzimy teraz podstawowe pojęcie — równoważności obiektów w systemie — z którego będziemy często korzystać w niniejszej książce.

W tym celu wprowadzimy dwie relacje dwuczłonowe w systemie informacyjnym $S=\langle X, A, V, \varrho\rangle$ w następujący sposób:

Definicja

Obiekty $x, y \in X$ są *nierozróżnialne* w systemie S *ze względu na atrybut* $a \in A$ (symbolicznie $x \underset{a}{\sim} y$) wtedy i tylko wtedy, gdy $\varrho_x(a)=\varrho_y(a)$. ∎

Definicja

Obiekty $x, y \in X$ są w systemie S *nierozróżnialne ze względu na każdy atrybut* $a \in A$ (albo krótko *nierozróżnialne* w systemie S); symbolicznie $x \underset{S}{\sim} y$ lub krótko $x \sim y$) wtedy i tylko wtedy, gdy $\varrho_x=\varrho_y$. ∎

Możemy też mówić o nierozróżnialności obiektów ze względu na pewien podzbiór $A' \subset A$ zbioru atrybutów, pisząc $x \underset{A'}{\sim} y$.

W systemie podanym w przykładzie 1.3 obiekty x_1, x_2 są nierozróżnialne ze względu na atrybut a, ponieważ $\varrho_{x_1}(a)=\varrho_{x_2}(a)$. Natomiast obiekty x_1, x_3 są nierozróżnialne ze względu na każdy atrybut systemu, ponieważ $\varrho_{x_1}=\varrho_{x_3}$.

Obie te relacje mają następującą ważną własność:

Własność 1.1

Dla każdego systemu informacyjnego $S=\langle X, A, V, \varrho\rangle$ relacje $\underset{a}{\sim}$ i $\underset{S}{\sim}$ są *relacjami równoważności* określonymi na zbiorze obiektów X i ponadto spełniają one warunek

$$\underset{S}{\sim} = \bigcap_{a\in A} \underset{a}{\sim}$$ ∎

Jest to ważna własność, która stanowi podstawę większości naszych dalszych rozważań — dlatego omówimy ją nieco dokładniej.

Każda relacja równoważności dzieli zbiór, na którym jest określona, na rozłączne klasy, które będziemy nazywali blokami tego podziału. Bloki podziału wyznaczonego przez relację $\underset{S}{\sim}$, nazwiemy *blokami* (albo *zbiorami*) *elementarnymi* systemu informacyjnego S.

Na przykład, jeżeli w systemie informacyjnym mamy atrybut KOLOR OCZU, to przy przyjętych przez nas założeniach dzieli on zbiór wszystkich obiektów na bloki zawierające osobników o tym samym kolorze oczu: blok zawierający wszystkie osoby z piwnymi oczyma, blok zawierający wszystkie osoby z niebieskimi oczyma, zielonymi itd. Atrybut ROK URODZENIA dzieli wszystkie obiekty systemu na bloki, z których każdy zawiera osobników urodzonych w tym samym roku. Zauważmy, że gdybyśmy dopuścili możliwość, że atrybut może mieć jednocześnie więcej niż jedną wartość, podział taki by nie istniał. Natomiast iloczyn podziałów wyznaczonych przez atrybuty KOLOR OCZU oraz ROK URODZENIA dzieli zbiór obiektów systemu na bloki, w których znajdują się obiekty o jednakowym kolorze oczu i roku urodzenia, np. do jednego bloku mogą należeć osoby urodzone w 1920 r. i posiadające piwne oczy; do innego — osoby urodzone w 1920 r. i mające niebieskie oczy, jeszcze do innego — osoby urodzone w 1930 r. i mające oczy niebieskie itp. Gdyby system ten zawierał tylko dwa wymienione atrybuty, tak otrzymane bloki byłyby blokami elementarnymi w tym systemie.

Następujący przykład ilustruje wprowadzone powyżej pojęcia.

Przykład 1.4

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym, w którym

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

$$A = \{a, b\}$$

$$V_a = \{p_1, p_2\}$$

$$V_b = \{q_1, q_2\}$$

funkcja ϱ zaś jest zdefiniowana tabelką

X	a	b
x_1	p_1	q_1
x_2	p_1	q_1
x_3	p_1	q_2
x_4	p_2	q_1
x_5	p_2	q_1
x_6	p_2	q_2

W przykładzie tym mamy następujące podziały:

Atrybut a dzieli zbiór X na bloki

$$B_1 = \{x_1, x_2, x_3\}$$

$$B_2 = \{x_4, x_5, x_6\}$$

Atrybut b dzieli zbiór X na bloki

$$B_3 = \{x_1, x_2, x_4, x_5\}$$

$$B_4 = \{x_3, x_6\}$$

Podział odpowiadający iloczynowi podziałów $\underset{a}{\sim}$ oraz $\underset{b}{\sim}$ składa się z bloków

$$B_5 = \{x_1, x_2\}$$

$$B_6 = \{x_4, x_5\}$$

$$B_7 = \{x_3\}$$

$$B_8 = \{x_6\}$$

Bloki B_5, B_6, B_7, B_8 są blokami elementarnymi. Sytuację tę przedstawiono poniżej. Bloki generowane przez poszczególne atrybuty zaznaczono liniami.

$$\underset{\underset{a}{\sim}}{\begin{array}{|ccc|} \hline x_1 & x_2 & x_3 \\ \hline x_4 & x_5 & x_6 \\ \hline \end{array}} \cap \underset{\underset{b}{\sim}}{\begin{array}{|cc|c|} \hline x_1 & x_2 & x_3 \\ & & \\ x_4 & x_5 & x_6 \\ \hline \end{array}} = \underset{\underset{a}{\sim} \cap \underset{b}{\sim}}{\begin{array}{|cc|c|} \hline x_1 & x_2 & x_3 \\ \hline x_4 & x_5 & x_6 \\ \hline \end{array}}$$ ■

Każdy system informacyjny wyznacza więc jednoznacznie pewien podział zbioru obiektów (a więc pewną ich klasyfikację) i odwrotnie, każda klasyfikacja obiektów wyznacza pewien system informacyjny. Jest to system z jednym atrybutem, którego wartościami mogą być na przykład liczby naturalne numerujące bloki elementarne systemu.

Podamy jeszcze jedną własność systemów informacyjnych, związaną z pojęciem informacji.

Niech φ, ψ będą różnymi informacjami w systemie S.
Wtedy

$$X_\varphi \cap X_\psi = \emptyset$$

$$\bigcup_{\varphi \in \mathrm{Inf}(S)} X_\varphi = X$$

Ponadto dla każdej informacji φ zbiór X_φ jest zbiorem elementarnym, więc informacje generują również podział (klasyfikację) zbioru obiektów; co więcej, podział ten jest identyczny z podziałem generowanym przez relację $\underset{S}{\sim}$. W ten sposób z każdym zbiorem elementarnym (niepustym) można jednoznacznie związać pewną informację w systemie S, i odwrotnie, z każdą niepustą informacją w systemie S można związać jednoznacznie pewien zbiór elementarny w systemie S. Informacje te możemy traktować jako opisy odpowiadających im zbiorów elementarnych (albo też ich nazwy).

Relacja równoważności $\underset{S}{\sim}$ będzie w dalszych naszych rozważaniach odgrywać zasadniczą rolę. Okaże się, że przy jej użyciu można określić wiele podstawowych pojęć związanych z systemem informacyjnym.

Definicja

Jeżeli systemy informacyjne S, S' mają ten sam zbiór obiektów, to powiemy, że S i S' są *równoważne* (symbolicznie $S \sim S'$) wtedy i tylko wtedy, gdy generują tę samą relację równoważności na zbiorze X, tj. $\underset{S}{\sim} = \underset{S'}{\sim}$. ■

Definicja

Jeżeli systemy informacyjne S, S' mają ten sam zbiór obiektów, to powiemy, że system S jest *dokładniejszy* od systemu S' wtedy i tylko wtedy, gdy $\underset{S}{\sim} \subset \underset{S'}{\sim}$. ■

Podział generowany przez system S jest „drobniejszy” od podziału generowanego przez system S', tzn. każdy blok elementarny systemu S' składa się z bloków elementarnych systemu S'.

Bliższą analizą tych pojęć zajmiemy się w dalszym ciągu tej książki po wprowadzeniu pojęcia języka systemu. Wprowadzimy jeszcze pojęcie reprezentacji systemu informacyjnego S.

Jeżeli w tablicy określającej system S opuścimy wszystkie wiersze powtarzające się, tj. zawierające tę samą informację, obiekty zaś zastąpimy w odpowiedniej kolumnie blokami zawierającymi te obiekty, to tak otrzymany system informacyjny nazwiemy reprezentacją systemu S i oznaczymy go przez S^*.

Definicja

Niech S będzie systemem informacyjnym, E_S zaś niech oznacza zbiór wszystkich bloków elementarnych systemu S.

Wprowadzimy funkcję $\varrho^*\colon E_S \times A \to V$ zdefiniowaną następująco:

$$\varrho^*(e, a) = v, \quad e \in E_S$$

wtedy i tylko wtedy, gdy istnieje takie $x \in e$, że

$$\varrho(x, a) = v$$

Jeżeli $S = \langle X, A, V, \varrho\rangle$ jest systemem informacyjnym, to system $S^* = \langle E_S, A, V, \varrho^*\rangle$ będziemy nazywać *reprezentacją systemu S.* ■

Łatwo stwierdzić, że podana definicja jest poprawna.

Przykład 1.5

Reprezentacją systemu podanego w przykładzie 1.4 o tablicy

X	a	b
x_1	p_1	q_1
x_2	p_1	q_1
x_3	p_1	q_2
x_4	p_2	q_1
x_5	p_2	q_1
x_6	p_2	q_2

jest system określony tabelką

E_S	a	b
$\{x_1, x_2\}$	p_1	q_1
$\{x_3\}$	p_1	q_2
$\{x_4, x_5\}$	p_2	q_1
$\{x_6\}$	p_2	p_2

Reprezentacja każdego systemu jest więc systemem selektywnym. W tablicy tego systemu każda informacja występuje tylko raz, a obiektami reprezentacji systemu S są bloki systemu S. ∎

Na zakończenie tego punktu zwrócimy uwagę na jeszcze jeden aspekt naszej definicji systemu informacyjnego, jednakże nie będziemy go w tej książce szerzej rozwijać.

Każdy system informacyjny S wyznacza pewną relację $R_S \subset V_{a_1} \times V_{a_2} \times \ldots \times V_{a_k}$, taką że $R_S(v_{i_1}, v_{i_2}, \ldots, v_{i_k})$ wtedy i tylko wtedy, gdy $(a_1, v_{i_1}), (a_2, v_{i_2}), \ldots, (a_k, v_{i_k})$ jest niepustą informacją w S, tj. gdy $X_{a_1, v_{i_1}} \cap X_{a_2, v_{i_2}} \cap \ldots \cap X_{a_k, v_{i_k}} \neq \emptyset$.

Znaczy to, że każdy system informacyjny można traktować jako pewną relację n-członową (gdy system ma n atrybutów). Kolejnymi dziedzinami tej relacji są zbiory wartości atrybutów systemu; ciąg wartości odpowiednich atrybutów należy do relacji, jeżeli istnieje obiekt w systemie, który ma własności wyrażone przez wartości atrybutów.

Poniższy przykład ilustruje podaną definicję.

NAZWISKO	ROK URODZENIA
Kowalski	1959
Michalski	1975
Kwiatkowski	1920

Relacja wyznaczona przez ten system jest dwuczłonowa (gdyż system ma tylko dwa atrybuty). Pierwszą dziedziną tej relacji jest zbiór wszystkich nazwisk, jakie mogą wystąpić w systemie (tutaj podaliśmy tylko trzy nazwiska) drugą dziedziną zaś są pewne liczby naturalne (daty urodzenia). Odpowiednia para NAZWISKO, DATA URODZENIA znajduje się w relacji wyznaczonej przez ten system wtedy, gdy istnieje w systemie osoba o danym nazwisku i dacie urodzenia.

Podejście takie, w którym system informacyjny traktujemy jako pewną relację, jest wielce użyteczne i zostało po raz pierwszy wprowadzone przez Codda [8]. Jest ono intensywnie badane przez wielu autorów (patrz np. prace [1], [16], [18]). My jednak nie będziemy się bliżej zajmować tym aspektem systemów informacyjnych. Zauważmy tylko, że definicja systemu relacyjnego jest konsekwencją przyjętej przez nas definicji systemu informacyjnego. Zauważmy również, że ściśle biorąc, nasza definicja systemu informacyjnego wyznacza nie jeden — lecz wiele systemów relacyjnych różniących się porządkiem argumentów relacji. Dla naszych celów nie jest sprawą istotną, w jakiej kolejności będziemy wymieniać argumenty relacji, np. jest sprawą obojętną, czy najpierw będziemy podawać NAZWISKO, a później ROK URODZENIA, czy też odwrotnie.

Pewne związki między systemem relacyjnym a atrybutowym, w sensie przyjętym w tej książce, były badane w pracy [27].

1.4. ZALEŻNOŚĆ ATRYBUTÓW

Często wartość jakiegoś atrybutu w systemie zależy od wartości innych atrybutów w tym systemie, tzn. jest ona jednoznacznie wyznaczona przez wartości innych atrybutów. Na przykład numer telefonu jest jednoznacznie wyznaczony przez nazwisko, imię oraz adres posiadacza telefonu. Numer kodu pocztowego jakiejś osoby jest jednoznacznie wyznaczony przez numer jej telefonu. Numer sali wykładowej jest jednoznacznie określony nazwą wykładu w rozkładzie zajęć. Podobnie nazwisko wykładowcy, dzień tygodnia i godzina jednoznacznie wyznacza wykład prowadzony przez tegoż wykładowcę. W systemie informacji medycznej adres pacjenta jednoznacznie wyznacza adres specjalistycznej przychodni lekarskiej, do której on należy. Mówiąc inaczej, nie wszystkie atrybuty grają w systemie jednakową rolę. Niektóre z nich mogą być wyeliminowane bez utraty informacji w systemie, gdyż ich wartości mogą być wyznaczone na podstawie pozostałych atrybutów.

Jeżeli wartość atrybutu a może być wyznaczona na podstawie wartości atrybutów $b_1, b_2, \ldots, b_k$ w systemie S, to powiemy, że atrybut a jest *zależny* od atrybutów $b_1, b_2, \ldots, b_k$ i zapiszemy $b_1, b_2, \ldots, b_k \underset{S}{\rightarrow} a$. Jeżeli system S jest ustalony i znany, to zamiast $\underset{S}{\rightarrow}$ będziemy pisać krócej $\rightarrow$, np.

NAZWISKO, IMIĘ, ADRES → NUMER TELEFONU

NUMER TELEFONU → KOD POCZTOWY

WYKŁAD → NUMER SALI

WYKŁADOWCA, DZIEŃ, GODZINA → WYKŁAD

Problem zależności atrybutów badało wielu autorów (np. Aho, Sagiv, Ullman [1], Armstrong [2], Fagin [10], Jaegermann, Marek [25], Orłowska [83]). W dotychczasowych badaniach przyjmowano, że zależności między atrybutami są z góry zadane. W przyjętym tu podejściu zależności między atrybutami w systemie nie są z góry zadane, jak to za chwilę zobaczymy, lecz są konsekwencją przyjętej definicji systemu informacyjnego, tj. wszystkie zależności między atrybutami w systemie są automatycznie zdefiniowane poprzez określenie systemu informacyjnego. Przyjmując więc zbiór obiektów atrybutów, ich wartości oraz funkcję ϱ, definiujemy automatycznie również wszystkie zależności między atrybutami.

Rozpatrzmy najpierw najprostszy przypadek zależności między dwoma atrybutami.

Niech a, b będą dwoma atrybutami w systemie informacyjnym $S = \langle X, A, V, \varrho \rangle$.

(a) Powiemy, że atrybut b *zależy od atrybutu* a (symbolicznie $a \rightarrow b$), gdy $\underset{a}{\sim} \subset \underset{b}{\sim}$;

(b) Atrybuty a, b są *niezależne w systemie* S, gdy nie zachodzi żadna z sytuacji $\underset{a}{\sim} \subset \underset{b}{\sim}$, $\underset{a}{\sim} \supset \underset{b}{\sim}$;

(c) Atrybuty a, b są *równoważne w systemie* S (symbolicznie $a \sim b$), gdy $\underset{a}{\sim} = \underset{b}{\sim}$.

Zauważmy, że jeżeli $a \rightarrow b$, to istnieje funkcja o argumentach w zbiorze wartości atrybutu a oraz przyjmująca wartości ze zbioru atrybutu b, jednoznacznie przyporządkowująca wartościom atrybutu a wartości atrybutu b. Dlatego w literaturze nazywa się tego typu zależności *zależnościami funkcjonalnymi*. Jeżeli atrybut b zależy od atrybutu a, to istnieje funkcja, taka że

$$f_a^b: V_a \to V_b$$
$$\varrho_x(b) = f_a^b(\varrho_x(a))$$

wtedy i tylko wtedy, gdy

$$X_{b,\varrho_x(b)} \supset X_{a,\varrho_x(a)}$$

Mając więc zadaną funkcję ϱ możemy dla każdej pary atrybutów a, b w systemie S sprawdzić, czy są one zależne czy też nie, i jeśli tak, uzyskać na podstawie funkcji ϱ funkcję f_a^b. Ponieważ rozpatrujemy tylko systemy o skończonych zbiorach obiektów, atrybutów i ich wartości, sprawdzenie czy dla dowolnych a, b zachodzi $a \to b$, polega na sprawdzeniu, czy w tabelce funkcji ϱ istnieją dwa wiersze o jednakowych wartościach w kolumnie a, lecz różnych w kolumnie b.

Przykład 1.6

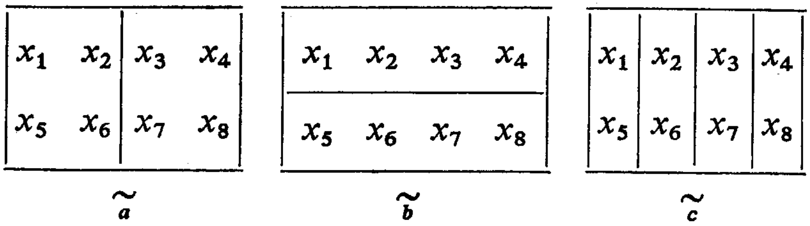
Rozpatrzmy system informacyjny $S = \langle X, A, V, \varrho \rangle$, w którym

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$
$$A = \{a, b, c\}$$
$$V_a = \{p_1, p_2\}$$
$$V_b = \{q_1, q_2\}$$
$$V_c = \{r_1, r_2, r_3, r_4\}$$

oraz funkcja ϱ jest określona tabelką

X	a	b	c
x_1	p_1	q_1	r_1
x_2	p_1	q_1	r_2
x_3	p_2	q_1	r_3
x_4	p_2	q_1	r_4
x_5	p_1	q_2	r_1
x_6	p_1	q_2	r_2
x_7	p_2	q_2	r_3
x_8	p_2	q_2	r_4

Z tablicy łatwo można zauważyć, że $c \to a$, natomiast a, b oraz c, b są parami niezależne. Możemy tę sytuację zilustrować, jak to pokazano poniżej:



Klasy równoważności (bloki) generowane przez poszczególne atrybuty zaznaczono liniami.

Ponieważ atrybut a zależy od atrybutu c, więc atrybut a jest w systemie tym zbędny, gdyż jego wartości można wyliczyć na podstawie wartości atrybutu c, znając oczywiście zależność zachodzącą między atrybutami c i a. Zależność ta w naszym przykładzie jest określona tabelką definiującą funkcję $f^a_c\colon V_c \to V_a$

a	c
p_1	r_1
p_1	r_2
p_2	r_3
p_2	r_4

Nasz system wyjściowy oraz system bez atrybutu a dają ten sam podział zbioru X, więc zgodnie z przyjętą poprzednio definicją równoważności systemów są to systemy równoważne. ■

W podobny sposób można zdefiniować zależność atrybutu a od podzbioru atrybutów B w systemie S, gdzie $B = \{b_1, b_2, \ldots, b_k\}$.

Atrybut *a zależy od podzbioru atrybutów* $B \subset A$ w systemie S (symbolicznie $B \to a$) wtedy i tylko wtedy, gdy $\underset{B}{\sim} \subset \underset{a}{\sim}$.

Jeżeli $B \to a$, to istnieje funkcja

$$f^a_B\colon V_{b_1} \times V_{b_2} \times \ldots \times V_{b_k} \to V_a$$

taka że

$$\varrho_x(a) = f^a_B(\varrho_x(b_1), \varrho_x(b_2), \ldots, \varrho_x(b_k))$$

wtedy i tylko wtedy, gdy

$$X_{a,\varrho_x(a)} \supset X_{b_1,\varrho_x(b_1)} \cap X_{b_2,\varrho_x(b_2)} \cap \ldots \cap X_{b_k,\varrho_x(b_k)}$$

dla każdego $a \in A$ oraz $x \in X$.

Przykład 1.7

Rozpatrzmy system informacyjny z następującymi zbiorami obiektów, atrybutów i wartości:

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$$

$$A = \{a, b, c\}$$

$$V_a = \{p_1, p_2, p_3, p_4\}$$

$$V_b = \{q_1, q_2, q_3\}$$

$$V_c = \{r_1, r_2, r_3\}$$

Funkcja ϱ dla tego systemu jest określona poniższą tabelką:

X	a	b	c
x_1	p_1	q_1	r_1
x_2	p_1	q_2	r_1
x_3	p_2	q_3	r_1
x_4	p_1	q_1	r_2
x_5	p_1	q_2	r_2
x_6	p_2	q_3	r_2
x_7	p_3	q_1	r_3
x_8	p_3	q_2	r_3
x_9	p_4	q_3	r_3

Atrybuty definiują więc w systemie następujące podziały:

$X_{a,p_1} = \{x_1, x_2, x_4, x_5\}$
$X_{a,p_2} = \{x_3, x_6\}$
$X_{a,p_3} = \{x_7, x_8\}$
$X_{a,p_4} = \{x_9\}$
$X_{b,q_1} = \{x_1, x_4, x_7\}$
$X_{b,q_2} = \{x_2, x_5, x_8\}$
$X_{b,q_3} = \{x_3, x_6, x_9\}$
$X_{c,r_1} = \{x_1, x_2, x_3\}$
$X_{c,r_2} = \{x_4, x_5, x_6\}$
$X_{c,r_3} = \{x_7, x_8, x_9\}$

Podziały te pokazano poniżej.

$$\underset{\tilde{a}}{\begin{array}{|cc|c|} \hline x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ \hline x_7 & x_8 & x_9 \\ \hline \end{array}} \qquad \underset{\tilde{b}}{\begin{array}{|c|c|c|} \hline x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \\ \hline \end{array}} \qquad \underset{\tilde{c}}{\begin{array}{|ccc|} \hline x_1 & x_2 & x_3 \\ \hline x_4 & x_5 & x_6 \\ \hline x_7 & x_8 & x_9 \\ \hline \end{array}}$$

W systemie tym atrybuty a, b i a, c są parami niezależne, natomiast atrybut a zależy od atrybutów b i c, tj. $\{b, c\} \to a$, lub pisząc krócej $b, c \to a$.

Odpowiednią funkcję $f_{b,c}^{a}$ przedstawiono poniżej.

a	b	c
p_1	q_1	r_1
p_1	q_1	r_2
p_3	q_1	r_3
p_1	q_2	r_1
p_1	q_2	r_2
p_3	q_2	r_3
p_2	q_3	r_1
p_2	q_3	r_2
p_4	q_3	r_3

Podobnie jak poprzednio, funkcję $f_{b,c}^{a}$ otrzymamy z tabelki funkcji ϱ dla rozpatrywanego systemu informacyjnego. ■

Możemy również rozpatrywać przypadek, w którym pewien zbiór atrybutów w systemie zależy od jakiegoś innego atrybutu, tj. $a \to B$, gdzie $B \subset A$ oraz $B = \{b_1, b_2, \ldots, b_k\}$.

W takim przypadku $a \to B$ wtedy i tylko wtedy, gdy $\underset{a}{\sim} \subset \underset{B}{\sim}$. Jeżeli $a \to B$, to istnieje układ funkcji $f_a^{b_1}, f_a^{b_2}, \ldots, f_a^{b_k}$, gdzie

$$f_a^{b_i}\colon V_a \to V_{b_i}$$

taki że

$$\varrho_x(b_i) = f_a^{b_i}(\varrho_x(a))$$

wtedy i tylko wtedy, gdy

$$X_{b_i, \varrho_x(b_i)} \supset X_{a, \varrho_x(a)}$$

dla każdego $x \in X$.

Wreszcie w ogólnym przypadku można wprowadzić zależność zbioru atrybutów C od zbioru atrybutów B, gdzie B, C są podzbiorami zbioru atrybutów A systemu S. Zbiór atrybutów C zależy od zbioru atrybutów B (symbolicznie $B \to C$) wtedy i tylko wtedy, gdy $\underset{B}{\sim} \subset \underset{C}{\sim}$.

Jeżeli $B \to C$, to istnieje układ funkcji

$$f_B^c\colon V_{b_1} \times V_{b_2} \times \ldots \times V_{b_k} \to V_c \quad \text{dla} \quad c \in C$$

taki że

$$\varrho_x(c) = f_B^c(\varrho_x(b_1), \varrho_x(b_2), \ldots, \varrho_x(b_k))$$

wtedy i tylko wtedy, gdy

$$X_{c, \varrho_x(c)} \supset X_{b_1, \varrho_x(b_1)} \cap X_{b_2, \varrho_x(b_2)} \cap \ldots \cap X_{b_k, \varrho_x(b_k)}$$

dla każdego $c \in C$ oraz $x \in X$.

Zależność ta jest oczywista i nie wymaga komentarza. Zauważmy jedynie, że sprawdzanie zależności między atrybutami systemu nie wymaga sprawdzania zawierania się odpowiednich zbiorów, co w przypadku maszynowej realizacji byłoby bardzo uciążliwe — lecz jedynie odpowiedniego sprawdzenia tablicy funkcji ϱ systemu, co jest znacznie prostsze.

Do sprawdzania zależności $B \to C$ w systemie S użyteczna będzie następująca prosta własność:

Własność 1.2

Jeżeli $B \to C$ w systemie S, to również $B \to C$ w reprezentacji S^* systemu S. ■

Zamiast więc sprawdzać zachodzenie zależności $B \to C$ w tablicy funkcji ϱ systemu S, można to sprawdzenie przeprowadzić w tablicy funkcji ϱ^* reprezentacji S^* systemu S. Tablica funkcji ϱ^* jest prostsza od tablicy funkcji ϱ (zawiera mniej wierszy), a więc sprawdzanie będzie przebiegało szybciej.

Przykład 1.8

Rozpatrzmy system informacyjny określony tablicą

X	a	b
x_1	v_1	w_1
x_2	v_1	w_1
x_3	v_1	w_2
x_4	v_1	w_2
x_5	v_2	w_3
x_6	v_2	w_3
x_7	v_2	w_4
x_8	v_2	w_4

Reprezentacja tego systemu ma postać

B	a	b
$\{x_1, x_2\}$	v_1	w_1
$\{x_3, x_4\}$	v_1	w_2
$\{x_5, x_6\}$	v_2	w_3
$\{x_7, x_8\}$	v_2	w_4

Z tablicy od razu widać, że $b \rightarrow a$. ■

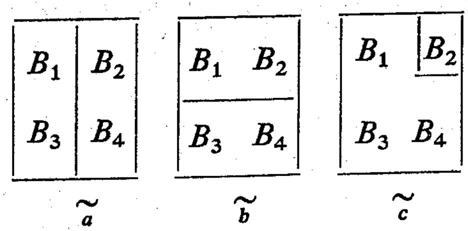
Wszystkie zależności funkcjonalne zachodzące w systemie informacyjnym można przedstawić w postaci grafu. Punktom grafu odpowiadają podzbiory atrybutów systemu informacyjnego; dwa punkty są w grafie połączone „strzałką", gdy odpowiednie podzbiory atrybutów są zależne. Ponieważ relacja równości atrybutów jest przechodnia, na grafie łączymy tylko bezpośrednio zależne podzbiory atrybutów. Jeżeli $B \rightarrow C$, to strzałkę na grafie kierujemy od B do C.

Przykład 1.9

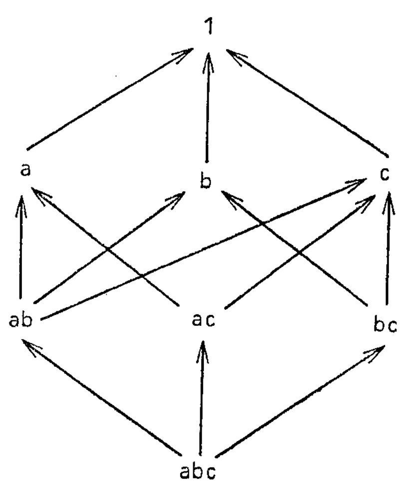
Rozpatrzymy wszystkie zależności funkcjonalne zachodzące w systemie określonym następującą tabelką:

X	a	b	c
B_1	v_1	w_1	u_1
B_2	v_2	w_1	u_2
B_3	v_1	w_2	u_1
B_4	v_2	w_2	u_1

Bloki tego systemu odpowiadające atrybutom a, b, c można przedstawić w sposób następujący:



Graf zależności funkcjonalnych będzie miał postać pokazaną na rys. 1.1.



Rys. 1.1. Graf zależności funkcjonalnych

Symbol 1 na rys. 1.1 oznacza podział zawierający tylko jeden blok, który jest równy całemu zbiorowi obiektów systemu. Podział ten nazywać będziemy *jednostkowym*. Podział jednostkowy zawiera oczywiście wszystkie podziały.

Zauważmy, że $a, b \to c$, gdyż $\underset{a}{\sim} \cap \underset{b}{\sim} \subset \underset{c}{\sim}$.

Podaliśmy poprzednio, że dla stwierdzenia, czy zachodzi funkcjonalna zależność w systemie S między atrybutami $B \to C$, wystarczy sprawdzić odpowiednie wiersze tablicy funkcji ϱ i w skończonej liczbie kroków otrzymamy odpowiedź, czy interesująca nas zależność funkcjonalna zachodzi czy też nie.

Dla niezbyt dużych tablic postępowanie takie jest praktycznie realizowalne, jednak przy bardzo dużych tablicach może ono być problematyczne z uwagi na zbyt długi czas potrzebny do zbadania całej tablicy dla stwierdzenia, czy dana funkcjonalna zależność zachodzi czy też nie. Nasuwa się pytanie, czy sprawy tej nie można rozwiązać w inny sposób. W przypadku ogólnym należałoby poddać pewne reguły wnioskowaniu, które pozwoliłoby na formalne dowodzenie zachodzenia lub nie zachodzenia badanych funkcjonalnych zależności.

Problem ten badało wielu autorów dla relacyjnych baz danych (patrz np. prace [1], [2], [10], [25], [83]).

1.5. SYSTEMY ZREDUKOWANE

Stwierdziliśmy w poprzednim punkcie, że niektóre atrybuty są w systemie zbędne, gdyż ich usunięcie z systemu nie zmienia bloków elementarnych w systemie. Przypominamy, że systemy, w których bloki elementarne są jednakowe, uważamy za równoważne. Oczywiście mogą istnieć powody, dla których usunięcie pewnych atrybutów z systemu nie jest wskazane, mimo że z punktu widzenia podziału na bloki elementarne są one zbędne. Do sprawy tej wrócimy jeszcze w dalszych rozdziałach, a obecnie atrybuty będziemy rozpatrywać tylko z punktu widzenia równoważności systemów.

W tym punkcie zajmiemy się sprawą „zbędnych" atrybutów w systemie informacyjnym, w podanym sensie. W tym celu wprowadzimy kilka potrzebnych definicji.

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym.

Definicja

Powiemy, że zbiór atrybutów $B \subset A$ jest *minimalny* w systemie S wtedy i tylko wtedy gdy dla każdego $B' \subset B$, $\underset{B}{\sim} \neq \underset{B'}{\sim}$. ■

Definicja

Zbiór $B \subset A$ jest *zależny* w systemie S wtedy i tylko wtedy, gdy istnieje taki zbiór atrybutów $B' \subset B$, że $\underset{B'}{\sim} = \underset{B}{\sim}$. ■

Definicja

Zbiór atrybutów A jest *wyprowadzalny* ze zbioru atrybutów B w systemie S wtedy i tylko wtedy, gdy $B \subset A$ oraz $\underset{A}{\sim} = \underset{B}{\sim}$. ■

Pierwsza definicja mówi, że zbiór atrybutów systemu jest minimalny, gdy usunięcie choć jednego z nich zmienia już podział zbioru obiektów na bloki elementarne.

Zbiór atrybutów jest zależny w systemie informacyjnym, jeżeli istnieją w tym zbiorze atrybuty takie, że ich usunięcie nie zmienia podziału zbioru obiektów systemu na bloki elementarne.

Wreszcie ostatnia definicja mówi, że zbiór atrybutów systemu jest wyprowadzalny z pewnego swego podzbioru atrybutów, gdy oba zbiory atrybutów generuje ten sam podział zbioru obiektów na bloki elementarne.

Uzasadnienie takiego właśnie rozumienia zależności, minimalności oraz wyprowadzalności atrybutów dają następujące własności.

Własność 1.3

Jeżeli zbiór atrybutów $B \subset A$ jest minimalny w systemie S, to dla każdego atrybutu $a \in A - B$, $B \rightarrow a$. ■

Własność 1.4

Jeżeli zbiór atrybutów B jest zależny w systemie S, to istnieje taki podzbiór $B' \subset B$, że $B \rightarrow a$ dla każdego atrybutu $a \in B - B'$. ■

Własność 1.5

Jeżeli $B \subset A$ oraz zbiór atrybutów A jest wyprowadzalny ze zbioru atrybutów B w systemie S, to $A \rightarrow A-B$. ■

Własność 1.6

Jeżeli zbiór atrybutów A jest minimalny w systemie S, to A nie jest wyprowadzalny z żadnego podzbioru właściwego A. ■

Własności te wskazują, że istotnie wprowadzone pojęcia zależności, niezależności oraz wyprowadzalności zbioru atrybutów są zgodne z intuicyjnym rozumieniem tych pojęć.

Wprowadzone pojęcia zilustrujemy przykładem.

Przykład 1.10

Rozpatrzmy system informacyjny określony tablicą

X	a	b	c	d
x_1	v_1	w_1	u_1	q_1
x_2	v_1	w_2	u_1	q_3
x_3	v_2	w_2	u_1	q_2
x_4	v_2	w_2	u_1	q_2
x_5	v_1	w_2	u_2	q_3

Atrybuty tego systemu generują następujące podziały:

$$\underset{a}{\sim} = \{x_1, x_2, x_5\}, \{x_3, x_4\}$$

$$\underset{b}{\sim} = \{x_1\}, \{x_2, x_3, x_4, x_5\}$$

$$\underset{c}{\sim} = \{x_1, x_2, x_3, x_4\}, \{x_5\}$$

$$\underset{d}{\sim} = \{x_1\}, \{x_3, x_4\}, \{x_2, x_5\}$$

Podział generowany przez cały system ma postać:

$$\underset{S}{\sim} = \{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5\}$$

W systemie tym mamy następujące zależności między atrybutami:

$$d \rightarrow b$$

$$d \rightarrow a$$

$$a, b, c \rightarrow d$$

$$c, d \rightarrow a, b$$

Zbiór atrybutów tego systemu jest więc zależny, gdyż zbiór atrybutów $\{a, b, c\}$ generuje taki sam podział na bloki elementarne jak, pełny system atrybutów. Podobnie zbiory atrybutów $\{a, c, d\}$ oraz $\{c, d\}$ mają taką samą własność, gdyż dają również taki sam podział na bloki elementarne zbioru obiektów, jak pełny zbiór atrybutów w systemie.

Zbiory atrybutów $\{a, b, c\}$ oraz $\{c, d\}$ są niezależne w systemie S, natomiast zbiór $\{a, c, d\}$ jest zależny w systemie S. ■

Wprowadzimy teraz podstawowe definicje tego punktu.

Niech $S = \langle X, A, V, \varrho\rangle$ będzie systemem informacyjnym.

Definicja

Zbiór atrybutów $B \subset A$ nazwiemy *reduktem zbioru atrybutów* A wtedy i tylko wtedy, gdy $\underset{B}{\sim} = \underset{A}{\sim}$ oraz nie istnieje właściwy podzbiór $B' \subset B$, takie że $\underset{B'}{\sim} = \underset{A}{\sim}$. Odpowiedni system $S' = \langle X, B, V', \varrho'\rangle$ nazwiemy *systemem zredukowanym* (funkcja ϱ' jest funkcją ϱ zredukowaną do zbioru $X \times B$, zaś $V' = \bigcup_{a \in B} V_a$).

Oczywiście system informacyjny może mieć więcej niż jeden redukt. System informacyjny podany w przykładzie 1.10 ma dwa redukty zbioru atrybutów — zbiory $\{a, b, c\}$ oraz $\{c, d\}$.

Oczywiście dla każdego systemu informacyjnego istnieje zredukowany system informacyjny jemu równoważny.

Ponadto, zachodzą następujące dwie własności:

Własność 1.7

Jeżeli system informacyjny jest kompletny, to jest także zredukowany (odwrotna własność nie zachodzi, patrz przykład 1.11). ■

Własność 1.8

Jeżeli system informacyjny jest zredukowany, to wszystkie jego różne atrybuty są parami niezależne (odwrotna własność nie zachodzi, patrz przykład 1.12). ■

Przykład 1.11

System informacyjny określony tabelką

X	a	b
x_1	v_1	w_1
x_2	v_2	w_1
x_3	v_1	w_2

jest zredukowany, lecz nie jest kompletny, gdyż brak w nim informacji (a, v_2), (b, w_2). ■

Przykład 1.12

W systemie informacyjnym określonym jak poniżej:

X	a	b	c
x_1	v_1	u_1	w_1
x_2	v_1	u_2	w_2
x_3	v_2	u_2	w_1
x_4	v_2	u_3	w_1

atrybuty a, b, c są parami niezależne, natomiast $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ są reduktami zbioru $\{a, b, c\}$ — a więc system ten nie jest zredukowany. ■

Często możliwość zredukowania systemu może mieć duże znaczenie praktyczne. Na przykład w przypadku diagnozy medycznej może się okazać, że dla uzyskania takich samych rezultatów wystarcza mniejszy zbiór atrybutów. Ma to w tym przypadku duże znaczenie, gdyż pozwala na posługiwanie się tylko tymi symptomami, które są istotnie niezbędne dla postawienia właściwej diagnozy.

Powstaje więc problem, jak znaleźć zredukowany system informacyjny dla zadanego systemu. Ponieważ rozważamy tylko systemy skończone, więc taki algorytm zawsze istnieje. Jest on jednakże bardzo nieefektywny [110], gdyż wymaga badania niezależności wszystkich podzbiorów zbioru atrybutów danego systemu. Byłoby rzeczą interesującą podanie algorytmu bardziej ekonomicznego, wykorzystującego np. pewne własności związane z pojęciem zależności atrybutów. Jednakże jak dotąd problem ten nie został rozwiązany. Pewne próby w tym kierunku można znaleźć w pracach [15], [25], [81].

Istnieje odpowiedniość między reduktami systemu informacyjnego S oraz reduktem reprezentacji S^*, dzięki czemu redukcje systemu można przeprowadzać na jego reprezentacji.

1.6. PODSYSTEMY INFORMACYJNE

Ważnym pojęciem będzie dla nas wprowadzone w tym punkcie pojęcie podsystemu informacyjnego.

Niech $S = \langle X, A, V, \varrho \rangle$ oraz $S' = \langle X', A', V', \varrho' \rangle$ będą dwoma systemami informacyjnymi.

Definicja

System S' jest *podsystemem* systemu S, gdy $X' \subset X$, $A' \subset A$ oraz $\varrho' = \varrho|_{X' \times A'}$. ■

Jeżeli S' jest podsystemem systemu S, to zapiszemy $S' < S$ lub $S' \underset{X', A'}{<} S$, bądź też $S' = S|_{X', A'}$.

Mówiąc prościej, jeżeli z tablicy funkcji ϱ systemu S usuniemy pewne wiersze lub kolumny, to tak otrzymana tablica przedstawia pewien podsystem systemu S.

Zauważmy, że dopuszczalne jest jednoczesne usunięcie zarówno wierszy, jak i kolumn w tablicy funkcji ϱ.

W ten sposób funkcja ϱ' wyznacza jednoznacznie zbiór V', taki że $S' = \langle X', A', V', \varrho' \rangle$ jest podsystemem systemu S.

Przykład 1.13

Jeżeli z tablicy definiującej system

X	a	b	c
x_1	v_1	u_2	w_1
x_2	v_2	u_1	w_2
x_3	v_1	u_2	w_2
x_4	v_1	u_2	w_1

usuniemy wiersz i kolumnę, jak to pokazano poniżej:

X	a	$\cancel{b}$	c
x_1	v_1	$\cancel{u_2}$	w_1
x_2	v_2	$\cancel{u_1}$	w_2
$\cancel{x_3}$	$\cancel{v_1}$	$\cancel{u_2}$	$\cancel{w_2}$
x_4	v_1	$\cancel{u_2}$	w_1

to otrzymamy podsystem

X	a	c
x_1	v_1	w_1
x_2	v_2	w_2
x_4	v_1	w_1

Odpowiada to sytuacji w rzeczywistym systemie, kiedy z systemu usuwamy pewne obiekty (np. książki z biblioteki) i jednocześnie pewne atrybuty (np. może przestać nas interesować, w jakim języku książkę wydano). ■

Zwróćmy uwagę, że zgodnie z przyjętą definicją podsystemu nie można dowolnie zmniejszyć zbioru wartości atrybutów w systemie, gdyż mogłoby to prowadzić do sytuacji, w której funkcja ϱ' nie byłaby wszędzie określona, co jest niedopuszczalne. Określoność funkcji ϱ' wymaga opuszczenia całego wiersza lub całej kolumny w tablicy systemu.

W dalszym ciągu będą nas szczególnie interesować dwa rodzaje podsystemów, a mianowicie takie, które otrzymuje się tylko przez usunięcie kolumn albo tylko przez usunięcie wierszy w tablicy funkcji ϱ, tzn. takie podsystemy, które otrzymuje się albo przez usunięcie pewnych cech, albo tylko przez usunięcie pewnych obiektów systemu.

Definicja

Jeżeli $S' < S$ oraz $X = X'$, to powiemy, że S' jest *podsystemem* systemu S *z ograniczonymi atrybutami* i zapiszemy $S' \underset{A'}{\leq} S$ lub $S' = S|_{A'}$. ■

Definicja

Jeżeli $S' < S$ oraz $A' = A$, to powiemy, że S' jest *podsystemem* systemu S *z ograniczonymi obiektami* i zapiszemy $S' \underset{X'}{\leq} S$ lub $S' = S|_{X'}$. ■

Przykład 1.14

Jeżeli w tablicy

X	a	b	c
x_1	v_1	u_2	w_1
x_2	v_2	u_1	w_2
x_3	v_1	u_2	w_2
x_4	v_1	u_2	w_1

opuścimy wiersz jak to pokazano poniżej:

X	a	b	c
x_1	v_1	u_2	w_1
x_2	v_2	u_1	w_2
~~x_3~~	~~v_1~~	~~u_2~~	~~w_2~~
x_4	v_1	u_2	w_1

to otrzymamy podsystem

X	a	b	c
x_1	v_1	u_2	w_1
x_2	v_2	u_1	w_2
x_4	v_1	u_2	w_1

z ograniczonymi obiektami.

Natomiast jeśli w tym systemie opuścimy kolumnę, jak to pokazano na poniższej tablicy,

X	a	~~b~~	c
x_1	v_1	~~u_2~~	w_1
x_2	v_2	~~u_2~~	w_2
x_3	v_1	~~u_2~~	w_2
x_4	v_1	~~u_2~~	w_1

to otrzymamy podsystem

X	a	c
x_1	v_1	w_1
x_2	v_2	w_2
x_3	v_1	w_2
x_4	v_1	w_1

z ograniczonymi atrybutami. ∎

Dla podsystemów danego systemu zachodzą dwie następujące, oczywiste własności.

Własność 1.9

Jeżeli S' jest podsystemem systemu S z ograniczonymi atrybutami, to $\underset{S'}{\sim} \supset \underset{S}{\sim}$. ∎

Własność 1.10

Jeżeli S' jest podsystemem systemu S z ograniczonymi obiektami, to $\underset{S'}{\sim} = \underset{S}{\sim} \cap (X')^2$. ∎

Ogólnie, usunięcie z systemu niezależnych atrybutów powoduje zmianę podziału obiektów systemu na bloki elementarne, i to w ten sposób, że nowych bloków jest mniej niż w oryginalnym systemie oraz są one sumą teoriomnogościową bloków systemu oryginalnego.

W przypadku utworzenia podsystemu danego systemu przez usunięcie z niego obiektów liczba bloków elementarnych w podsystemie jest nie większa niż w systemie oryginalnym i ich wielkość jest taka sama lub mniejsza niż w systemie oryginalnym.

Tak więc usunięcie atrybutów niezależnych w systemie prowadzi do zmniejszenia liczby bloków elementarnych i zwiększenia ich wielkości, natomiast usunięcie obiektów w systemie prowadzi ewentualnie do zmniejszenia liczby bloków elementarnych w systemie lub do zmniejszenia ich wielkości.

Przykład 1.15

Rozpatrywany w poprzednim przykładzie system miał trzy bloki elementarne $\{x_1, x_4\}$, $\{x_2\}$, $\{x_3\}$, natomiast w podsystemie otrzymanym przez usunięcie obiektu x_3 otrzymamy bloki elementarne $\{x_1, x_4\}$, $\{x_2\}$, w podsystemie zaś otrzymanym przez usunięcie atrybutu b są trzy elementarne: $\{x_1, x_4\}$, $\{x_2\}$, $\{x_3\}$, gdyż usunięty atrybut b jest zależny od atrybutów a i c. Gdybyśmy w systemie tym usunęli atrybut c, wówczas otrzymany podsystem posiadałby dwa bloki elementarne $-\{x_1, x_2, x_4\}$ i $\{x_3\}$.

Podamy jeszcze kilka elementarnych własności podsystemów danego systemu. ∎

Własność 1.11

Jeżeli S' jest dowolnym podsystemem systemu S oraz system S jest zredukowany, to system S' jest również zredukowany. ■

Jeżeli S' jest podsystemem systemu S z ograniczonymi obiektami oraz S jest kompletny, to system S' nie musi być kompletny.

Własność 1.12

Jeżeli S' jest podsystemem systemu S z ograniczonymi atrybutami oraz S jest kompletny, to również system S' jest kompletny. ■

Jeżeli S' jest podsystemem systemu S z ograniczonymi atrybutami oraz S jest selektywny, to S' nie musi być selektywny.

Własność 1.13

Jeżeli S' jest podsystemem systemu S z ograniczonymi obiektami oraz S jest selektywny, to również S' jest selektywny. ■

Własność 1.14

Jeżeli system S' jest podsystemem systemu S z ograniczonymi obiektami, to reprezentacja systemu S' jest podsystemem z ograniczonymi obiektami reprezentacji systemu S. ■

Jeżeli system S' jest podsystemem systemu S z ograniczonymi atrybutami, to reprezentacja systemu S' może nie być podsystemem reprezentacji systemu S.

Własność 1.15

Jeżeli S' jest dowolnym podsystemem systemu S, tj. $S' = S|_{Y,B}$, to istnieją systemy $S_1 = S|_Y$ oraz $S_2 = S|_B$, takie że $S' = S_1|_B = S_2|_Y$, przy czym systemy S_1 i S_2 są jedyne. ■

Podane własności pozwolą nam lepiej się zorientować, jak wpływa usuwanie obiektów bądź atrybutów z systemu na jego charakter.

1.7. ŁĄCZENIE SYSTEMÓW INFORMACYJNYCH

W tym punkcie omówimy ważną operację na systemach informacyjnych, a mianowicie ich łączenie. Rozważmy najpierw prosty przykład łączenia systemów informacyjnych. Załóżmy, że mamy katalogi biblioteczne dla dwu różnych bibliotek. Obie biblioteki zawierają różne książki, choć niektóre mogą występować w obu bibliotekach, oraz różne zbiory atrybutów, za pomocą których książki te są opisywane. Oczywiście niektóre atrybuty mogą być użyte w obu bibliotekach.

Jeżeli biblioteki te chcemy połączyć z jakichś powodów w jedną, powstaje pytanie, jak należy stworzyć katalog tak powstałej biblioteki. Czy jest on po prostu sumą obu katalogów, czy też są tu konieczne inne operacje? Przykład ten daje dobrą intuicję dalszych rozważań prowadzonych w tym punkcie.

Niech dane będą dwa systemy informacyjne $S' = \langle X', A', V', \varrho' \rangle$ oraz $S'' = \langle X'', A'', V'', \varrho'' \rangle$.

Definicja

System $S = \langle X, A, V, \varrho \rangle$ jest *złączeniem* systemów S' oraz S'' (symbolicznie $S = S' \cup S''$), gdy są spełnione następujące warunki:

$$X = X' \cup X''$$

$$A = A' \cup A''$$

$$V = V' \cup V''$$

$$\varrho|_{X' \times A'} = \varrho'$$

$$\varrho|_{X'' \times A''} = \varrho''$$

$$\varrho_x = \varrho'_x \cup \varrho''_x \quad \text{dla } x \in X.$$

Warunki te mówią, że w systemie złączonym obiekty, atrybuty i wartości atrybutów są sumami obiektów, atrybutów i wartości atrybutów systemów składowych, a informacja o obiektach jest sumą informacji o obiektach w systemach składowych.

Aby złączenie systemów było dobrze określone, muszą być spełnione dwa następujące warunki (*warunki zgodności*):

(1) Jeżeli $X' \cap X'' \neq \emptyset$ oraz $A' \cap A'' \neq \emptyset$, to

$$\varrho'|_{(X' \cap X'') \times (A' \cap A'')} = \varrho''|_{(X' \cap X'') \times (A' \cap A'')}$$

oraz

(2) Dla każdego $x \in X$ funkcja ϱ_x jest określona dla każdego atrybutu $a \in A$.

Pierwszy warunek jest oczywisty. Mówi on tyle, że jeżeli w obu systemach składowych występują jednakowe obiekty i jednakowe atrybuty, to informacja o tych obiektach w obu systemach musi być jednakowa. Jest to naturalny warunek i nie wymaga wyjaśnienia.

Drugi warunek, równie oczywisty, wymaga jednak krótkiego omówienia. Mówi on to, że jeżeli jakiś atrybut występuje w jednym ze składowych systemów, to musi również występować w drugim składowym systemie; w przeciwnym przypadku w wyniku złożenia nie otrzymalibyśmy systemu informacyjnego.

Jeżeli bowiem atrybut a występuje tylko na przykład w systemie S', natomiast nie występuje w systemie S'', to w rezultacie dla obiektów należących do systemu S'' nie można znaleźć wartości tego atrybutu — a w konsekwencji nie można określić informacji o dowolnym obiekcie w systemie złożonym.

Na przykład gdy system S' jest systemem informacji medycznej, S'' zaś systemem informacji ubezpieczeniowej, to jest sprawą oczywistą, że oba te systemy mają różne zbiory atrybutów. Aby po ich złączeniu był spełniony warunek, że funkcja ϱ w systemie złączonym jest wszędzie określona, konieczne jest posiadanie informacji ubezpieczeniowej i medycznej o każdym obiekcie w systemie złączonym.

Przykład 1.16

Rozpatrzmy złączenie dwu następujących systemów:

X	a	b	c
x_1	v_1	u_1	w_2
x_2	v_1	u_2	w_1
x_3	v_2	u_1	w_2
x_4	v_2	u_1	w_2

Y	c	d	e
x_3	w_2	p_1	q_1
x_4	w_2	p_2	q_1
y_1	w_2	p_3	q_1
y_2	w_2	p_1	q_2

Złączenie obu tych systemów określa następująca tabelka:

$X \cup Y$	a	b	c	d	e
x_1	v_1	u_1	w_2	—	—
x_2	v_1	u_2	w_1	—	—
x_3	v_2	u_1	w_2	p_1	q_1
x_4	v_2	u_1	w_2	p_2	q_1
y_1	—	—	w_1	p_3	q_1
y_2	—	—	w_2	p_1	q_2

Złączenie to nie jest dobrze określone, gdyż np. nie są znane wartości atrybutów a, b dla obiektów y_1, y_2 oraz wartości atrybutów d, e dla obiektów x_1, x_2. ■

A więc złączenia takiego poprawnie wykonać nie można. Praktycznie interesujący jest jedynie przypadek poprawnego złączenia dwu różnych systemów, gdy mają one jednakowe atrybuty bądź jednakowe obiekty. Gdy zbiory atrybutów oraz zbiory obiektów w obu systemach składowych są różne i żaden z systemów składowych nie jest swoim właściwym podsystemem, wówczas złączenie takich systemów nie jest systemem informacyjnym.

Dlatego też wprowadzimy dwa rodzaje złączenia systemów, które nazwiemy złączeniem atrybutami oraz złączeniem obiektami.

Definicja

Jeżeli $S' = \langle X, A', V', \varrho' \rangle$ oraz $S'' = \langle X, A'', V'', \varrho'' \rangle$, to ich złączenie $S = S' \cup S''$, nazwiemy *złączeniem atrybutami*. ■

Definicja

Jeżeli $S' = \langle X', A', V', \varrho' \rangle$ oraz $S'' = \langle X'', A', V', \varrho'' \rangle$, to ich złączenie $S = S' \cup S''$ nazwiemy *złączeniem obiektami*. ■

Oba rodzaje złączeń są oczywiście dobrze określone i tylko takimi dwoma rodzajami będziemy zajmować się w tej książce.

Przykład 1.17

Złączeniem systemów

X	a	b	c
x_1	u_1	v_1	w_2
x_2	u_1	v_2	w_1
x_3	u_2	v_1	w_2
x_4	u_1	v_1	w_2

X	a	d	e
x_1	u_1	p_1	q_2
x_2	u_1	p_2	q_1
x_3	u_2	p_1	q_1
x_4	u_1	p_1	q_2

jest system określony tabelką

X	a	b	c	d	e
x_1	u_1	v_1	w_2	p_1	q_2
x_2	u_1	v_2	w_1	p_2	q_1
x_3	u_2	v_1	w_2	p_1	q_2
x_4	u_1	v_1	w_2	p_2	q_2

Natomiast złączeniem systemów

X	a	b	c
x_1	u_1	v_1	w_2
x_2	u_2	v_2	w_1
x_3	u_1	v_2	w_1
x_4	u_1	v_1	w_1

Y	a	b	c
x_3	u_1	v_2	w_1
x_4	u_1	v_1	w_1
y_1	u_2	v_2	w_2
y_2	u_1	v_2	w_1

jest system określony tabelką

$X \cup Y$	a	b	c
x_1	u_1	v_1	w_2
x_2	u_2	v_2	w_1
x_3	u_1	v_2	w_1
x_4	u_1	v_1	w_1
y_1	u_2	v_2	w_2
y_2	u_1	v_2	w_1

W pierwszym przypadku mamy oczywiście do czynienia ze złączeniem atrybutami, w drugim zaś — ze złączeniem obiektami. ■

Jeżeli mamy dwa systemy informacyjne, np. medyczny i ubezpieczeniowy, dotyczące tych samych ludzi, to możemy je złączyć atrybutami, natomiast jeżeli mamy np. dwa systemy medyczne znajdujące się w różnych miastach, to możemy je złączyć obiektami, tworząc w ten sposób jeden system informacyjny.

Łączyć można nie tylko dwa systemy informacyjne, lecz także dowolną ich liczbę.

Definicja

Niech $S_1, S_2, \ldots, S_n$ będą systemami informacyjnymi i niech przy tym $S_i = \langle X_i, A_i, V_i, \varrho^i \rangle$. *Złączeniem systemów* $S_1, S_2, \ldots, S_n$ będziemy nazywali system $S = \langle X, A, V, \varrho \rangle$, taki że

$$X = \bigcup_{i=1}^{n} X_i$$

$$A = \bigcup_{i=1}^{n} A_i$$

$$V = \bigcup_{i=1}^{n} V_i$$

$$\varrho|_{X_i \times A_i} = \varrho^i \quad \text{dla } i = 1, \ldots, n,$$

$$\varrho_x = \bigcup_{i=1}^{n} \varrho_x^i \quad \text{dla } x \in X$$

■

Złączenie jest dobrze określone, gdy są spełnione następujące warunki:

(1) Jeżeli $X_i \cap X_j \neq \emptyset$ oraz $A_i \cap A_j \neq \emptyset$, to

$$\varrho^i|_{(X_i \cap X_j) \times (A_i \cap A_j)} = \varrho^j|_{(X_i \cap X_j) \times (A_i \cap A_j)}$$

dla wszystkich $i, j = 1, 2, \ldots, n$.

(2) Dla każdego $x \in X$

$$\varrho_x = \bigcup_{i=1}^{n} \varrho_x^i$$

jest określona dla wszystkich $a \in A$.

Zauważmy, że operacja łączenia systemów informacyjnych nie jest w ogólnym przypadku łączna, tzn. nie zachodzi warunek

$$S_1 \cup (S_2 \cup S_3) = (S_1 \cup S_2) \cup S_3$$

Własność tę ilustruje poniższy przykład.

Przykład 1.18

Niech S_1, S_2, S_3 będą określone tabelami jak to pokazano poniżej.

System S_1

X	a	b	c
x_1	u_1	v_1	w_1
x_2	u_2	v_1	w_1
x_3	u_1	v_1	w_2
x_4	u_1	v_2	w_2
x_5	u_2	v_2	w_1
x_6	u_1	v_1	w_2

System S_2

X	c	d	e
x_1	w_1	p_1	q_1
x_2	w_1	p_1	q_1
x_3	w_2	p_2	q_1
x_4	w_2	p_1	q_2

System S_3

X	c	d	e
x_3	w_2	p_2	q_1
x_4	w_2	p_1	q_2
x_5	w_1	p_2	q_1
x_6	w_1	p_1	q_1

Złączenie obiektami systemów S_2 i S_3 jest dopuszczalne i będzie miało postać

X	c	d	e
x_1	w_1	p_1	q_1
x_2	w_1	p_1	q_1
x_3	w_2	p_2	q_1
x_4	w_2	p_1	q_2
x_5	w_1	p_2	q_1
x_6	w_1	p_1	q_1

System $S_2 \cup S_3$ możemy teraz połączyć obiektami z systemem S_1 i otrzymamy w rezultacie system $S_1 \cup (S_2 \cup S_3)$ określony tabelką

X	a	b	c	d	e
x_1	u_1	v_1	w_1	p_1	q_1
x_2	u_2	v_1	w_1	p_1	q_1
x_3	u_1	v_1	w_2	p_2	q_1
x_4	u_1	v_2	w_2	p_1	q_2
x_5	u_2	v_2	w_1	p_2	q_1
x_6	u_1	v_1	w_2	p_1	q_1

Gdybyśmy chcieli złączyć najpierw systemy S_1 i S_2, a następnie system $S_1 \cup S_2$ złączyć z systemem S_3, wówczas działanie to byłoby z formalnego punktu widzenia nieokreślone. ■

Łączność zachodzi w przypadku, gdy $S_i = S|_{A_i}$ bądź $S_i = S|_{X_i}$ dla każdego $i = 1, 2, \ldots, k$. Podamy teraz kilka prostych własności operacji łączenia systemów. Niech $S = \langle X, A, V, \varrho \rangle$ oraz $S_i = \langle X_i, A_i, V_i, \varrho^i \rangle$.

Własność 1.16

Jeżeli $S = \bigcup_{i=1}^{k} S_i$, to oczywiście każdy system S_i jest podsystemem systemu S. ■

Własność 1.17

Jeżeli $S = \bigcup_{i=1}^{k} S_i$, $S_i = S|_{X_i}$ oraz każdy system S_i jest zredukowany, to również system S jest zredukowany. ■

Jeżeli $S = \bigcup_{i=1}^{k} S_i$, $S_i = S|_{A_i}$ oraz każdy system S_i jest zredukowany, to system S nie musi być zredukowany.

Własność 1.18

Jeżeli system $S = \bigcup_{i=1}^{k} S_i, S_i = S|_{A_i}$, to $\underset{S}{\sim} = \bigcap_{i=1}^{k} \underset{S_i}{\sim}$. ■

Znaczy to, że bloki relacji równoważności $\underset{S}{\sim}$ są iloczynami teoriomnogościowymi odpowiednich bloków relacji systemów składowych.

Każdy system informacyjny można więc traktować jako złączenie atrybutami pewnych jego podsystemów z ograniczonymi atrybutami. W szczególnym przypadku każdy system informacyjny jest złączeniem atrybutami wszystkich jego jednoatrybutowych podsystemów z ograniczonymi atrybutami.

Jeżeli system informacyjny $S = \bigcup_{i=1}^{k} S_i, S_i = S|_{X_i}$, to każdy blok relacji równoważności $\underset{S}{\sim}$ jest sumą teoriomnogościową pewnych bloków relacji $\underset{S_i}{\sim}$.

Ściślej, jeżeli φ jest informacją w systemie S, to

$$X_\varphi = \bigcup_{i=1}^{k} X_{i,\varphi}$$

dla każdego $\varphi \in \mathrm{Inf}(S)$, gdzie $X_{i,\varphi}$ oznacza blok elementarny w systemie S_i odpowiadający informacji φ.

Własność 1.19

Jeżeli systemy $S = \bigcup_{i=1}^{k} S_i, S_i = S|_{X_i}$ oraz S_i są zredukowane, to również S jest zredukowany. ■

Jeżeli system $S = \bigcup_{i=1}^{k} S_i, S_i = S|_{A_i}$ oraz systemy S_i są zredukowane, to system S nie musi być zredukowany.

Własność 1.20

Jeżeli $S = \bigcup_{i=1}^{k} S_i$ oraz $S_i = S|_{X_i}$, to $S^* \neq (\bigcup_{i=1}^{k} S_i)^*$. ■

Własność 1.21

Jeżeli $S = \bigcup_{i=1}^{k} S_i$ oraz $S_i = S|_{A_i}$, to $S^* = (\bigcup_{i=1}^{k} S_i)^*$. ■

Własność 1.22

Jeżeli $S = \bigcup_{i=1}^{k} S_i$, $S_i = S|_{A_i}$ oraz systemy S_i są selektywne, to system S jest selektywny. ■

Jeżeli $S = \bigcup_{i=1}^{k} S_i$, $S_i = S|_{X_i}$ oraz systemy S_i są selektywne, to S nie musi być systemem selektywnym.

Własność 1.23

Jeżeli $S = \bigcup_{i=1}^{k} S_i$ oraz S_i są systemami kompletnymi, to S jest systemem kompletnym. ■

Reprezentacja złączenia $S = \bigcup_{i=1}^{k} S_i$ nie zawsze jest złączeniem reprezentacji S_i, tj. istnieją takie systemy S_i, że

$$(\bigcup_{i=1}^{k} S_i)^* \neq \bigcup_{i=1}^{k} (S_i)^*$$

2. JĘZYKI SYSTEMÓW INFORMACYJNYCH

2.1. WPROWADZENIE

W poprzednim rozdziale wprowadziliśmy pojęcie systemu informacyjnego. Obecnie możemy przystąpić do analizy języka systemu informacyjnego, tj. języka, w którym będziemy formułować pytania dotyczące informacji zawartych w systemie. Język taki jest często nazywany *językiem zapytań.* Zanim jednak przejdziemy do bardziej szczegółowego omawiania języka, zastanówmy się przez chwilę nad rodzajem pytań, jakie mogą być stawiane przez użytkowników systemu informacyjnego, a przy okazji poświęcimy nieco uwagi problemowi: co to jest pytanie i odpowiedź. Problem ten był od dawna analizowany przez logików, my jednakże będziemy tutaj patrzyli na to zagadnienie z nieco innego punktu widzenia niż logicy — a mianowicie, będziemy tu patrzeć na język zapytań z punktu widzenia zastosowań — w jak najbardziej efektywny sposób uzyskać odpowiedź na postawione pytanie w danym systemie informacyjnym.

Rozpatrzmy najpierw prosty system biblioteczny. Najczęściej spotykane pytanie w takim systemie brzmi:

Podać książkę napisaną przez X pod tytułem Y wydaną w roku Z.

Odpowiedzią na takie pytanie jest właśnie numer katalogowy żądanej książki lub książek — gdy w bibliotece istnieje kilka egzemplarzy tej samej książki. Odpowiedzią jest więc pewien zbiór obiektów należących do systemu.
Czasem możemy spotkać się z pytaniami postaci:

Podać wszystkie książki z dziedziny X, wydane po roku Y w języku Z.

Odpowiedzią na takie pytanie będzie również zbiór wszystkich obiektów systemu spełniających warunki wymienione w pytaniu — tj. zbiór numerów katalogowych wszystkich książek z dziedziny X wydanych po roku Y w języku Z, znajdujących się w danej bibliotece.

Można również oczekiwać pytań następującej postaci:

Podać wszystkie książki z dziedziny X w językach Z_1, Z_2, Z_3 z wyjątkiem książek autora Y.

Ktoś może już mieć wszystkie książki autora Y i nie jest zainteresowany otrzymaniem informacji na ten temat. Cechą charakterystyczną wszystkich pytań wymienionych poprzednio jest to, że odpowiedzią na nie jest zbiór pewnych obiektów systemu, a mianowicie tych wszystkich obiektów, które spełniają warunki podane w pytaniu. Pytania takie będziemy w dalszym ciągu nazywać *pytaniami mnogościowymi.*

Ściśle biorąc, wyrażenia postaci „Podaj wszystkie książki itd...” nie są z językowego punktu widzenia pytaniami, a raczej żądaniami (i tak są często nazywane w literaturze dotyczącej systemów informacyjnych) — my jednak pozostaniemy przy terminie „pytanie”, z uwagi na ujednolicenie terminologii. Wiele rodzajów żądań, którymi zajmiemy się w dalszym ciągu ma bowiem postać pytań.

Rozpatrzmy obecnie drugą ważną grupę pytań. Najlepiej je zilustrować na przykładzie systemu informacyjnego, którego obiektami są ludzie. W takim przypadku możemy być zainteresowani nie tylko pewnymi zbiorami obiektów systemu, lecz także pewnymi związkami zachodzącymi między nimi, np. możemy być zainteresowani wszystkimi parami małżeńskimi bądź wszystkimi dziećmi danej osoby (lub pary małżeńskiej), bądź też spadkobiercami danej osoby.

Pytania mogą mieć w takim przypadku postać:

Podaj wszystkie pary X, Y, takie że X, Y są małżeństwem (lub prościej — podaj wszystkie małżeństwa w systemie).

Podaj wszystkich spadkobierców X.

Podaj wszystkie dzieci X.

Podaj wszystkich przodków X.

Odpowiedzią na pierwsze pytanie jest pewien zbiór par obiektów systemu — a więc pewna relacja dwuczłonowa. Natomiast odpowiedziami na pozostałe trzy pytania są podzbiory obiektów systemu, różnią się one jednak istotnie od odpowiedzi otrzymywanych w pytaniach mnogościowych, a mianowicie elementy tych zbiorów są w określonym związku z innym wyróżnionym obiektem systemu, a więc w istocie odpowiedzi te są wyznaczone przez pewne relacje, dlatego pytania te zaliczamy do tej samej grupy. Pytania dotyczące związków między obiektami będziemy nazywać *pytaniami relacyjnymi*. Jest to ważna grupa pytań, występuje ona w wielu systemach informacyjnych, np. podstawową rolę pytania relacyjne odgrywają w systemach informacji patentowej. Jesteśmy w takim przypadku bowiem często zainteresowani znalezieniem wszystkich możliwych powiązań jakiegoś patentu z innymi patentami.

Następną grupę będą stanowiły pytania, które nazwiemy *liczbowymi*. Mogą one mieć na przykład postać:

Ile jest osób znających język angielski?

Ile jest publikacji na temat X napisanych po roku Y?

Ile małżeństw zawarto w roku X?

We wszystkich tego rodzaju pytaniach pytamy o liczbę dokumentów pewnych zbiorów lub relacji. W pytaniach mnogościowych bądź relacyjnych interesowały nas same zbiory bądź relacje, natomiast w pytaniach liczbowych pytamy o liczbę elementów tych zbiorów czy też relacji. Pytanie tego rodzaju może być często zadawane przed odpowiednim pytaniem mnogościowym bądź relacyjnym. Pytający może być bowiem zainteresowany uzyskaniem najpierw odpowiedzi dotyczącej liczności szukanego zbioru.

Dalszą ważną grupę będą stanowiły *pytania numeryczne*. Przykładami takich pytań są:

Podać maksymalny wiek pracowników zatrudnionych w zakładzie X.
Podać średnie wynagrodzenie pracowników wydziału X w roku Y.
Podać dochód X w roku Y.
Podać koszty produkcji wyrobu X w zakładzie Y.
Podać ilość zużytego materiału X do produkcji wyrobu Y na wydziale Z.

Cechą charakterystyczną tej grupy pytań jest to, że dla uzyskania odpowiedzi musimy wykonać pewne obliczenia numeryczne na wartościach niektórych cech obiektów — stąd właśnie nazwa, numeryczne.

Zwróćmy uwagę na różnicę między pytaniami liczbowymi a numerycznymi. W pierwszych i drugich pytaniach odpowiedzią jest liczba: w pytaniach liczbowych odpowiedź dotyczyła liczności pewnych zbiorów i była uzyskiwana przez zliczenie liczby elementów pewnych zbiorów obiektów. W pytaniach numerycznych odpowiedź jest uzyskana w wyniku obliczeń otrzymanych na wartościach niektórych cech obiektów. Sposób uzyskania odpowiedzi jest więc w obu przypadkach całkiem różny, a to właśnie, jak podano na początku tego punktu jest podstawą wydzielenia odpowiednich grup pytań.

Wreszcie ostatnią rozpatrywaną tu grupą będą *pytania logiczne*. Odpowiedzią na pytania logiczne jest wartość prawda lub fałsz, stąd właśnie nazwa, lub po prostu odpowiedź może mieć w tym przypadku postać TAK lub NIE.

Na przykład, często spotkać się możemy z pytaniami logicznymi postaci:

Czy X jest równe Y?
Czy X ma wzrost 175 cm?
Czy X jest autorem książki Y?
Czy średnia dochodów rocznych w zakładzie X w roku Y wynosi Z?

Dla uzyskania odpowiedzi na pytania tego rodzaju należy sprawdzić, czy odpowiednie relacje zachodzą między wskazanymi obiektami. Można to uczynić bądź przez przegląd odpowiednich obiektów i sprawdzenie, czy mają one żądaną własność czy też nie, bądź też metodą odpowiednich rozumowań. Ta druga metoda jest o wiele ciekawsza i efektywniejsza, jednakże nie zawsze można się nią posługiwać. Sprawami tymi zajmiemy się w dalszym ciągu.

Podana tu analiza pytań nie jest wyczerpująca. Istnieje wiele innych rodzajów pytań, które są interesujące z rozpatrywanego punktu widzenia. Byłoby rzeczą interesującą dokładniejsze zbadanie tej sprawy. Tutaj chcielibyśmy jedynie wyróżnić naszym zdaniem najważniejsze z praktycznego punktu widzenia grupy pytań oraz zasygnalizować konieczność bardziej szczegółowego zajęcia się tym problemem, który, o ile mi wiadomo, nie był do tej pory badany.

Reasumując, będziemy rózróżniali następujące rodzaje pytań:
mnogościowe,
relacyjne,

liczbowe,
numeryczne,
logiczne.

W konsekwencji, przystępując do konstruowania języka zapytań powinniśmy najpierw ustalić, jakiego rodzaju pytania chcemy zadawać w naszym systemie. Język taki może obejmować jednocześnie różne rodzaje pytań, może też dotyczyć tylko jednego rodzaju pytań. Ten drugi przypadek jest znacznie prostszy z formalnego punktu widzenia i dlatego w dalszym ciągu zajmiemy się tylko takimi właśnie językami. Języki te będziemy nazywali odpowiednio:

mnogościowymi,
relacyjnymi,
liczbowymi,
numerycznymi,
logicznymi.

W dalszym ciągu omówimy szczegółowo najpierw pierwszy język, język mnogościowy i na jego przykładzie przedyskutujemy główne problemy występujące w związku z konstruowaniem języków zapytań systemów informacyjnych.

2.2. SKŁADNIA JĘZYKA

Jak to zapowiedzieliśmy w poprzednim punkcie, rozpatrzymy obecnie dokładniej język mnogościowy, tj. język zapytań, w którym odpowiedziami na pytania są pewne zbiory obiektów systemu. Język systemu informacyjnego S będziemy oznaczać przez L_S.

Najpierw określimy składnię języka L_S, tj. określimy reguły, według których będziemy budować wyrażenia poprawne w języku L_S — zwane dalej *termami.*

Termy są zbudowane z symboli *alfabetu,* do którego należą:

(1) deskryptory systemu, tj. pary symboli postaci (a, v), gdzie $a \in A$, $v \in V_a$;

(2) stałe 0, 1;

(3) symbole operacji $\sim, \cdot, +, \to, \leftrightarrow$, zwane odpowiednio: negacją, koniunkcją, alternatywą, implikacją i równoważnością. Symbole operacji można uważać za skróty spójników logicznych „nie", „i", „lub", „jeżeli... to", „wtedy i tylko wtedy".

Zauważmy, że w języku L_S nie ma zmiennych, a jedynie występują w nim stałe oraz symbole operacji. Ponadto zwróćmy uwagę, że zbiory atrybutów A oraz wartości atrybutów V_a traktujemy jako zbiory pewnych nazw, tj. np. atrybut KOLOR utożsamiany z tymże słowem. W języku angielskim użylibyśmy tu słowa COLOUR, a więc ten sam zbiór atrybutów w języku polskim i angielskim miałby różną postać. Podobnie przedstawia się sprawa z wartościami atrybutów. W języku polskim moglibyśmy mieć wartość atrybutu KOLOR — „czerwony", natomiast w języku angielskim tej samej wartości odpowiadałoby słowo „red".

Dla ścisłości więc należałoby rozróżniać atrybuty od ich nazw i podobnie wartości atrybutów od ich nazw w konkretnym języku naturalnym. Dla celów niniejszej pracy rozróżnienie takie nie jest jednak konieczne i dlatego nie będziemy takiego rozróżnienia czynili, utożsamiając atrybuty z ich nazwami i podobnie utożsamiając wartości atrybutów z ich nazwami.

Wyrażeniami poprawnymi, czyli *termami* języka L_S będą dowolne stałe i deskryptory połączone symbolami operacji.

Zbiór termów języka L_S określimy więc następująco:

(1) Stałe 0, 1 są termami;

(2) Deskryptory systemu S, tj. pary stałych (a, v), $a \in A$, $v \in V_a$ są termami;

(3) Jeżeli t, t' są termami, to również termami są

$$\sim t \qquad (t \to t')$$
$$(t + t') \qquad (t \leftrightarrow t')$$
$$(t \cdot t')$$

W termach będziemy używać nawiasów zgodnie z przyjętymi zwyczajami.

Przykłady termów podano poniżej.

(NAZWISKO = Kowalski)
(PŁEĆ = mężczyzna) · (WIEK = 17)
(ZAWÓD = księgowy) + (WYNAGRODZENIE = 6000 zł)
0 + (PŁEĆ = kobieta)
1 · (WAGA = 76 kg)
~ [(KOLOR = czerwony) · (WIELKOŚĆ = mały)]
(NAZWISKO = Malinowski) → (WYNAGRODZENIE = 5000 zł)
(OCZY = niebieskie) ↔ (WŁOSY = blond)

Uwaga: W przykładach termów, zgodnie ze zwyczajem przyjętym w językach programowania, będziemy między atrybutem a jego wartością pisać zamiast przecinka znak równości. Ponadto atrybuty będziemy pisać dużymi literami, ich wartości zaś małymi.

W podanych przykładach zakładamy oczywiście, że występujące w nich atrybuty i ich wartości należą do alfabetu rozpatrywanego języka.

Przypominamy, że termy są pytaniami. Nie wszystkie z pytań, które można utworzyć w naszym języku, są interesujące z praktycznego punktu widzenia. Nie potrafimy jednak odróżnić formalnie pytań interesujących od nie interesujących i dlatego w dalszym ciągu będziemy jako pytania traktować wszystkie termy języka, niezależnie od tego, jaki to ma sens praktyczny.

Komentarza wymagają również stałe 0, 1. Są one potrzebne jedynie z formalnego punktu widzenia, po to abyśmy mogli dokonywać na pytaniach operacji formalnych.

2.3. SEMANTYKA JĘZYKA

W poprzednim punkcie określono, jak są zbudowane wyrażenia poprawne (termy) języka systemu informacyjnego, obecnie określimy ich znaczenie. Znaczeniem termu, zgodnie z naszą intencją, ma być pewien podzbiór zbioru obiektów, który stanowi odpowiedź na pytanie reprezentowane przez term. Znaczenie termów w systemie S jest więc określone przez funkcję σ_S, której argumentami są termy języka L_S, wartościami zaś podzbiory zbioru obiektów systemu S. Funkcję tę będziemy nazywać *semantyką języka* L_S. Gdy system S jest ustalony, zamiast σ_S będziemy pisać σ.

Znaczenie termów w systemie S jest określone indukcyjnie ze względu na złożoność termu w następujący sposób:

(1) $\sigma_S(0) = \emptyset, \qquad \sigma_S(1) = X$

(2) $\sigma_S(a, v) = \{x \in X: \varrho_x(a) = v\}$

(3) $\sigma_S(\sim t) = X - \sigma_S(t)$

(4) $\sigma_S(t+t') = \sigma_S(t) \cup \sigma_S(t')$

(5) $\sigma_S(t \cdot t') = \sigma_S(t) \cap \sigma_S(t')$

(6) $\sigma_S(t \to t') = \sim \sigma_S(t) \cup \sigma_S(t')$

(7) $\sigma_S(t \leftrightarrow t') = \sigma_S(t \to t') \cap \sigma_S(t' \to t)$

Z podanej definicji widać jasno, w jaki sposób rozumiemy odpowiedź na pytanie.

Odpowiedzią na pytanie reprezentowane przez stałą 0 jest zbiór pusty. Odpowiedzią na pytanie reprezentowane przez term 1 jest cały zbiór obiektów systemu. Stałą 1 można więc traktować jako pytanie o wszystkie obiekty w systemie. Oczywiście pytanie takie nie jest praktycznie ciekawe. Pytania reprezentowane przez stałe 0, 1 mają charakter formalny i będą używane jedynie dla uproszczenia termów.

Odpowiedź na pytanie reprezentowane przez deskryptor postaci (a, v) jest zbiorem wszystkich obiektów systemu, mających własność wyrażoną przez deskryptor (a, v). Na przykład jeżeli deskryptor ma postać

(OCZY = zielone)

to odpowiedzią na takie pytanie systemu jest zbiór wszystkich osób mających zielone oczy, oczywiście osób, o których informacje znajdują się w rozpatrywanym przez nas systemie.

Odpowiedzią na zaprzeczenie pytania jest zbiór wszystkich obiektów systemu nie posiadających własności wyrażonych w pytaniu. Na przykład odpowiedzią na pytanie

$\sim$ (OCZY = zielone)

będzie zbiór osób posiadających oczy o kolorze różnym od zielonego.

Odpowiedzią na pytanie będące sumą dwóch pytań, będzie suma termomnogościowa odpowiedzi na pytania składowe. Na przykład, jeżeli pytanie ma postać

$$(\text{OCZY} = \text{zielone}) + (\text{WZROST} = 170)$$

odpowiedzią będzie zbiór osób, które mają zielone oczy lub 170 cm wzrostu.

Wreszcie odpowiedzią na pytanie będące iloczynem pytań jest zbiór obiektów mających jednocześnie własności wyrażone w obu pytaniach. Na przykład odpowiedzią na pytanie:

$$(\text{OCZY} = \text{zielone}) \cdot (\text{WZROST} = 170)$$

jest zbiór osób, które jednocześnie mają zielone oczy oraz 170 cm wzrostu.

Szczególnie interesujące są pytania zawierające implikację i równoważność. Na przykład odpowiedzią na pytanie

$$(\text{STANOWISKO} = \text{mistrz}) \to (\text{WYNAGRODZENIE} = 5000 \text{ zł})$$

zgodnie z określeniem znaczenia implikacji

$$\sim (\text{STANOWISKO} = \text{mistrz}) + (\text{WYNAGRODZENIE} = 5000 \text{ zł})$$

będzie zbiór tych osób, które nie mają stanowiska mistrza lub mają wynagrodzenie 5000 zł.

Zwróćmy uwagę, że jeżeli wszyscy mistrzowie figurujący w rozpatrywanym przez nas systemie mają wynagrodzenie 5000 zł, to odpowiedzią na to pytanie będzie zbiór wszystkich obiektów naszego systemu, w przeciwnym przypadku natomiast, odpowiedzią będzie zbiór wszystkich obiektów systemu z wyjątkiem tych, które są mistrzami i jednocześnie mają wynagrodzenie różne od 5000 zł, tzn. jeżeli

$$\sigma_S\,[(\text{STANOWISKO} = \text{mistrz}) \to (\text{WYNAGRODZENIE} = 5000 \text{ zł})] = X$$

to $\sigma_S\,(\text{STANOWISKO} = \text{mistrz}) \subset \sigma_S\,(\text{WYNAGRODZENIE} = 5000 \text{ zł})$

Znaczenie termu zawierającego równoważność możemy określić w podobny sposób. Na przykład znaczeniem termu

$$(\text{OCZY} = \text{niebieskie}) \leftrightarrow (\text{WŁOSY} = \text{blond})$$

będzie zbiór tych wszystkich osób figurujących w rozpatrywanym systemie z wyjątkiem tych, które mają oczy niebieskie i włosy różne od blond lub — włosy blond i jednocześnie oczy różne od niebieskich.

Jeżeli odpowiedzią jest zbiór wszystkich obiektów systemu S, tzn.

$$\sigma_S\,[(\text{OCZY} = \text{niebieskie}) \leftrightarrow (\text{WŁOSY} = \text{blond})] = X$$

to zbiory osób mających niebieskie oczy oraz blond włosy są identyczne, tj.

$$\sigma_S\,(\text{OCZY} = \text{niebieskie}) = \sigma_S\,(\text{WŁOSY} = \text{blond})$$

Jakkolwiek więc odpowiedzią na ostatnie dwa rodzaje pytań są zbiory obiektów, to jednakże w istocie pytania te mają charakter logiczny, a nie teoriomnogościowy, gdyż odpowiedzią na nie są wyrażenia TAK lub NIE.

W pierwszym przykładzie pytanie

$$(\text{STANOWISKO} = \text{mistrz}) \rightarrow (\text{WYNAGRODZENIE} = 5000\ \text{zł})$$

należy rozumieć jako pytanie „Czy każdy mistrz (w danym systemie informacyjnym) ma wynagrodzenie 5000 zł?

Jeżeli

$$\sigma_S\,[(\text{STANOWISKO} = \text{mistrz}) \rightarrow (\text{WYNAGRODZENIE} = 5000\ \text{zł})] = X$$

to odpowiedzią jest TAK, w przeciwnym przypadku — NIE.

Podobnie sytuacja wygląda dla pytań zawierających równoważność. W istocie pytanie:

$$(\text{OCZY} = \text{niebieskie}) \leftrightarrow (\text{WŁOSY} = \text{blond})$$

jest pytaniem postaci

„Czy zbiór osób mających niebieskie oczy jest identyczny ze zbiorem osób mających blond włosy?".

Odpowiedzią na takie pytanie jest wyrażenie TAK lub NIE. Odpowiedź TAK otrzymamy, gdy

$$\sigma_S\,[(\text{OCZY} = \text{niebieskie}) \leftrightarrow (\text{WŁOSY} = \text{blond})] = X$$

tj.

$$\sigma_S\,(\text{OCZY} = \text{niebieskie}) = \sigma_S\,(\text{WŁOSY} = \text{blond})$$

W przeciwnym przypadku odpowiedzią na to pytanie jest NIE. Jakkolwiek sprawa pytania i odpowiedzi jest dla rozpatrywanej klasy języków oczywista, zrobimy na zakończenie tego punktu jeszcze prosty przykład formalny ilustrujący pojęcie termu i jego znaczenia.

Przykład 2.1

Niech S będzie następującym systemem informacyjnym:

X	a	b	c
x_1	v_1	w_1	u_2
x_2	v_2	w_1	u_3
x_3	v_1	w_2	u_1
x_4	v_1	w_2	u_1
x_5	v_2	w_2	u_3
x_6	v_1	w_1	u_3

Jako alfabet języka tego systemu przyjmiemy oprócz stałych 0, 1, symboli operacji $\sim, \cdot, +, \rightarrow, \leftrightarrow$ również atrybuty a, b, c oraz wartości $v_1, v_2, w_1, w_2, u_1, u_2, u_3$. W naszym języku termami są następujące wyrażenia:

$(a, v_1)+(b, w_2)\cdot(c, u_2)$

$\sim[(a, v_2)\cdot(a, v_1)]+(c, u_3)$

$(b, w_1)+(c, u_1)$

$(b, w_1)\rightarrow(c, u_3)$

$(a, v_2)\leftrightarrow(b, w_2)$

Znaczeniem tych termów w systemie S są następujące zbiory:

$$\sigma_S((a, v_1)+(b, w_2)\cdot(c, u_2)) = \{x_1, x_3, x_4, x_6\}\cup(\{x_3, x_4, x_5\}\cap\{x_1\}) =$$

$$= \{x_1, x_3, x_4, x_6\}$$

$$\sigma_S(\sim[(a, v_2)\cdot(a, v_1)]+(c, u_3)) = \sim(\emptyset)\cup\{x_2, x_5, x_6\} = X$$

$$\sigma_S((b, w_1)+(c, u_1)) = \{x_1, x_2, x_6\}\cup\{x_3, x_4\} = \{x_1, x_2, x_3, x_4, x_6\}$$

$$\sigma_S((b, w_1)\rightarrow(c, u_3)) = (X-\{x_1, x_2, x_6\})\cap\{x_2, x_5, x_6\} = \{x_2, x_3, x_4, x_5, x_6\}$$

$$\sigma_S((a, v_2)\leftrightarrow(b, w_2)) = \{x_1, x_3, x_4, x_5, x_6\}\cap\{x_1, x_2, x_5, x_6\} = \{x_1, x_5, x_6\}$$ ■

Definicja

Powiemy, że term t jest *prawdziwy* w systemie S (symbolicznie $\models_S t$), gdy $\sigma_S(t) = X$. ■

Własność 2.1

Jeżeli term $t\rightarrow t'$ jest prawdziwy w systemie S, to $\sigma_S(t)\subset\sigma_S(t')$. ■

Własność 2.2

Jeżeli term $t\leftrightarrow t'$ jest prawdziwy w systemie S, to $\sigma_S(t) = \sigma_S(t')$. ■

W dalszym ciągu będziemy używali następujących notacji: Zamiast $\models_S t\rightarrow t'$ będziemy pisać też $t\underset{S}{\leqslant} t'$ (lub $t\leqslant t'$, gdy system S jest znany) oraz zamiast $\models_S t\leftrightarrow t'$ zapiszemy $t\underset{S}{=} t'$ (lub $t = t'$, gdy system S jest znany).

Definicja

Jeżeli $t\underset{S}{\leqslant} t'$, to powiemy, że term t jest *dokładniejszy* od termu t' w systemie S. ■

Definicja

Jeżeli $t\underset{S}{=} t'$, to powiemy, że termy t i t' są *równe semantycznie* w systemie S. ■

Na zakończenie tego punktu wprowadzimy jeszcze dwa szczególne rodzaje termów potrzebne nam w dalszym ciągu, a mianowicie pojęcie termu prostego w systemie S oraz pojęcie termu charakterystycznego systemu S.

Definicja

Powiemy, że term t języka L_S jest *prosty*, gdy t jest postaci

$$(a_1, v_1)\cdot(a_2, v_2)\cdot\ldots\cdot(a_n, v_n)$$

gdzie $a_1, a_2, \ldots, a_n$ są wszystkimi atrybutami (różnymi) ze zbioru A oraz $v_1, v_2, \ldots,$ $\ldots, v_n$ są pewnymi wartościami odpowiednio ze zbiorów $V_{a_1}, V_{a_2}, \ldots, V_{a_n}$. ■

Termy proste mają następującą ważną własność:

Własność 2.3

Jeżeli t jest termem prostym w języku L_S, to $\sigma_S(t)$ jest zbiorem elementarnym w systemie S lub zbiorem pustym. ■

Tak więc termy proste stanowią opisy klas równoważności relacji S (zbiorów elementarnych).

Przykład 2.2

Dla systemu S określonego tabelką

X	a	b
x_1	v_1	u_2
x_2	v_2	u_3
x_3	v_1	u_2
x_4	v_2	u_1

termami prostymi w języku L_S są

$$(a, v_1) \cdot (b, u_1)$$
$$(a, v_1) \cdot (b, u_2)$$
$$(a, v_1) \cdot (b, u_3)$$
$$(a, v_2) \cdot (b, u_1)$$
$$(a, v_2) \cdot (b, u_2)$$
$$(a, v_2) \cdot (b, u_3)$$

Wartości tych termów są następujące:

$$\sigma_S((a, v_1) \cdot (b, u_1)) = \emptyset$$
$$\sigma_S((a, v_1) \cdot (b, u_2)) = \{x_1, x_3\}$$
$$\sigma_S((a, v_1) \cdot (b, u_3)) = \emptyset$$
$$\sigma_S((a, v_2) \cdot (b, u_1)) = \{x_4\}$$
$$\sigma_S((a, v_2) \cdot (b, u_2)) = \emptyset$$
$$\sigma_S((a, v_2) \cdot (b, u_3)) = \{x_2\}$$

Zbiorami elementarnymi w systemie S są

$$\{x_1, x_3\}, \{x_2\}, \{x_4\}$$ ■

Drugim ważnym termem jest term charakterystyczny systemu informacyjnego.

Definicja

Powiemy, że term $t \in L_S$ jest *pusty w systemie S*, jeżeli $\sigma_S(t) = 0$. ■

Definicja

Niech $t_1, t_2, \ldots, t_m$, $m \geqslant 1$, będą wszystkimi termami prostymi niepustymi systemu S. Term $t_S = t_1 + t_2 + t_3 + \ldots + t_m$ będziemy nazywać termem *charakterystycznym* systemu S. ■

Na przykład termem charakterystycznym systemu S rozpatrywanego w poprzednim przykładzie jest

$$t_S = (a, v_1) \cdot (b, u_2) + (a, v_2) \cdot (b, u_3) + (a, v_2) \cdot (b, u_1)$$

Term charakterystyczny systemu S ma następującą ważną własność:

Własność 2.4

Jeżeli term t_S jest termem charakterystycznym systemu S, to t_S jest prawdziwy w systemie S, tj.

$$\sigma_S(t_S) = X$$ ■

Podkreślmy jeszcze raz, że znaczenie termu zależy od systemu informacyjnego, w którym ten term ma określoną semantykę, a więc np. pytanie dotyczące wszystkich osób o nazwisku Kowalski nie dotyczy wszystkich Kowalskich istniejących w rzeczywistości, lecz tylko tych, którzy figurują w konkretnym, interesującym nas aktualnie systemie informacyjnym. W szczególności term t może być prawdziwy w jednym systemie, a nieprawdziwy w innym z tymi samymi zbiorami X, A, V. Mogą istnieć jednak takie termy, których prawdziwość nie zależy od konkretnych wartości występujących w nich atrybutów, a jedynie od ich budowy formalnej. Będą one grały szczególną rolę w naszych rozważaniach. Zwróćmy uwagę, że do uzyskania odpowiedzi na taki term nie jest potrzebna znajomość wiedzy zawartej w jakimkolwiek systemie informacyjnym. Odpowiedź na nie wynika bowiem z ich struktury logicznej. A więc, jeżeli pytamy czy jakiś fakt zachodzi w świetle naszej aktualnej wiedzy (reprezentowanej w danym systemie informacyjnym) wystarczy w odpowiedni sposób skorzystać z tej wiedzy, natomiast w przypadku pytań o fakty prawdziwe w każdym możliwym systemie informacyjnym nie jest potrzebny w istocie żaden system informacyjny, lecz odpowiednie środki logiczne.

2.4. REGUŁY PRZEKSZTAŁCANIA TERMÓW

Język wprowadzony w poprzednich punktach był związany z konkretnym systemem informacyjnym. Ten sam język można traktować jako język wspólny dla wielu systemów informacyjnych o różnych zbiorach obiektów, ale jednakowych zbiorach atrybutów i jednakowych zbiorach wartości atrybutów. Od strony składni (tj. syntaktycznej) wszystkie języki takich systemów będą jednakowe, natomiast różnić się będzie ich semantyka z uwagi na różne zbiory obiektów w systemach informacyjnych, w których określamy znaczenie termów.

Dla sprawdzania równości semantycznej termów konieczne są odpowiednie reguły transformacji termów, które przekształcają termy, zachowując ich znaczenie. Oczywiście moglibyśmy również sprawdzić równość semantyczną termów, obliczając na podstawie definicji ich znaczenia i porównując, czy są one jednakowe czy też nie. Takie postępowanie byłoby jednakże wysoce niepraktyczne, chociaż — ze względu na skończoność rozpatrywanych systemów informacyjnych — zawsze możliwe. Sprawdzanie tą metodą równości termów w każdym systemie informacyjnym nie byłoby już możliwe z uwagi na konieczność obliczania znaczenia termów w nieskończonej liczbie systemów informacyjnych. Dlatego też są tu konieczne odpowiednie aksjomaty pozwalające sprawdzić, czy też dowodzić, równości termów w sposób formalny. W tym celu przyjmiemy odpowiednie równości, z których część jest po prostu zmodyfikowanymi nieco prawami algebry Boole'a, część zaś jest związana ze specyfiką systemów informacyjnych. Pierwsza część aksjomatów składająca się z trzech grup A, B, C może być traktowana jako reguły przekształcania termów.

Niech t, p, s będą termami.

A1. $(t+p)+s=t+(p+s)$ B1. $(t\cdot p)\cdot s=t\cdot(p\cdot s)$

A2. $t+p=p+t$ B2. $t\cdot p=p\cdot t$

A3. $t\cdot(p+s)=t\cdot p+t\cdot s$ B3. $t+(p\cdot s)=(t+p)\cdot(t+s)$

A4. $\sim(t+p)=\sim t\cdot\sim p$ B4. $\sim(t\cdot p)=\sim t+\sim p$

A5. $t+t=t$ B5. $t\cdot t=t$

A6. $t+0=t$ B6. $t\cdot 0=0$

A7. $t+(t\cdot s)=t$ B7. $t\cdot(t+s)=t$

A8. $\sim 0=1$ B8. $\sim 1=0$

A9. $t+1=1$ B9. $t\cdot 1=\mathrm{t}$

A10. $t+\sim t=1$ B10. $t\cdot\sim t=0$

A11. $\sim(\sim t)=t$

C1. $t\to p=\sim t+p$

C2. $t\leftrightarrow p=t\cdot p+\sim t\cdot\sim p$

C3. $t\leftrightarrow p=\sim t\leftrightarrow\sim p$

Podane powyżej równości mają charakter ogólny i zachodzą w każdym systemie informacyjnym. Ponadto, w każdym konkretnym systemie informacyjnym S zachodzą następujące prawa:

D1. $(a,v)\cdot(a,u)=0,\ u,v\in V_a,\ u\neq v,$ dla każdego $a\in A$;

D2. $\sum\limits_{v\in V_a}(a,v)=1,$ dla każdego $a\in A$;

D3. $\sim(a, v) = \sum_{u \in V_a} (a, u),\ u \neq v$, dla każdego $a \in A$;

D4. $t_S = 1$, t_S — term charakterystyczny systemu S.

Aksjomaty grup A, B, C są prawami algebry Boole'a, natomiast aksjomaty grupy D, jakkolwiek oczywiste, wymagają krótkiego komentarza. Są one bowiem specyficzne dla naszego pojęcia systemu informacyjnego.

Aksjomat D1 wynika z przyjętego w poprzednim rozdziale założenia, że informacja ϱ_x o obiekcie x jest funkcją, tzn. przyjęliśmy, że każdy obiekt może mieć dokładnie jedną wartość każdego atrybutu. Na przykład jeżeli coś ma kolor czerwony, to tym samym nie ma koloru niebieskiego ani zielonego.

Aksjomat D2 wynika z założenia o pełności informacji, tzn. każdy obiekt musi przyjmować jedną z wartości każdego atrybutu. Na przykład jeżeli atrybutem jest KOLOR, to każdy obiekt musi mieć jakiś kolor, który jest wartością tego atrybutu.

Aksjomat D3 pozwala usuwać negację w ten sposób, że zamiast mówić, iż obiekt nie ma jakiejś własności, można powiedzieć, że ma on jedną z pozostałych własności, np. zamiast mówić, że coś nie jest czerwone, można powiedzieć, że jest ono zielone lub niebieskie, lub fioletowe itd. Oczywiście reguła ta jest dopuszczalna z uwagi na skończoność zbioru wartości każdego atrybutu.

Aksjomat D4 wyraża następujący fakt: „$t_S = 1$" oznacza, iż wykluczamy z naszego systemu elementy, których opisy nie występują jako składniki termu charakterystycznego systemu S.

Niech t, p będą termami języka L_S. Jeżeli term t może być otrzymany z termu p za pomocą reguł A, B, C, D, to termy t i p są *semantycznie równe* w systemie informacyjnym S; odwrotnie, jeżeli termy t i p języka L_S są semantycznie równe w systemie informacyjnym S, to każdy z nich można otrzymać z pozostałego za pomocą reguł A, B, C, D.

Własność taka jest nazywana *pełnością języka*. Pozwala ona na w pełni „bezpieczne" posługiwanie się regułami A, B, C, D, gwarantując, że stosowanie tych reguł zachowuje znaczenie termów. Znaczy to, że nasz język zapytań jest zbudowany prawidłowo i możemy się nim posługiwać tak, jak każdym innym językiem formalnym stosowanym w matematyce.

2.5. POSTAĆ NORMALNA TERMÓW

Reguły podane w poprzednim punkcie pozwalają przekształcić każdy term w term mu równoważny, posiadający specjalną postać, zwaną postacią normalną. Postać ta ma duże znaczenie dla dalszych naszych rozważań. Pozwala ona każde pytanie przedstawić w jednolitej postaci, umożliwiającej wygodny sposób znalezienia odpowiedzi.

Celem zdefiniowania postaci normalnej termu przypomnijmy najpierw pojęcie termu prostego. Definicję termu prostego podano w poprzednim punkcie, nie będziemy więc tu jej powtarzać, a podamy jedynie kilka przykładów termów prostych.

Termami prostymi są na przykład:

(PŁEĆ = mężczyzna) · (WYNAGRODZENIE = duże) · (WIEK = młody)

(NAZWISKO = Kowalski) · (ZAWÓD = nauczyciel) · (OCZY = niebieskie)

Przykład 2.3

W systemie informacyjnym

X	a	b	c
x_1	v_1	u_1	w_2
x_2	v_2	u_1	w_1
x_3	v_1	u_1	w_2
x_4	v_1	u_2	w_2
x_5	v_2	u_1	w_1
x_6	v_2	u_1	w_1

mamy trzy zbiory elementarne $\{x_1, x_3\}$, $\{x_2, x_5, x_6\}$, $\{x_4\}$, którym odpowiadają termy proste

$$(a, v_1) \cdot (b, u_1) \cdot (c, w_2)$$

$$(a, v_2) \cdot (b, u_1) \cdot (c, w_1)$$

$$(a, v_1) \cdot (b, u_2) \cdot (c, w_2)$$

■

Definicja

Powiemy, że term t jest *w postaci normalnej* wtedy i tylko wtedy, gdy $t = 0$, $t = 1$ lub $t = t_1 + t_2 + \ldots + t_k$, gdzie t_i są termami prostymi. ■

Termy mają następującą ważną własność:

Własność 2.5

Dla każdego termu t istnieje term t' w postaci normalnej równoważny t. ■

Przykład 2.4

Sprowadźmy do postaci normalnej następujący term:

$$t_1 = \sim [(a, v_1) \cdot (b, u_2)] + (b, u_1)$$

przyjmując, że rozpatrywany język ma atrybuty i wartości takie, jak w systemie informacyjnym omawianym w przykładzie 2.3, tj. atrybutami są a, b, c, zbiory ich wartości zaś są następujące:

$$V_a = \{v_1, v_2\}$$

$$V_b = \{u_1, u_2\}$$

$$V_c = \{w_1, w_2\}$$

Na podstawie reguły B4 można term t_1 przedstawić w postaci

$$t_2 = \sim(a, v_1) + \sim(b, u_2) + (b, u_1)$$

skąd na podstawie reguły D3 otrzymamy

$$t_3 = (a, v_2) + (b, u_1) + (b, u_1)$$

Stosując do t_3 regułę A5, otrzymamy

$$t_4 = (a, v_2) + (b, u_1)$$

Zgodnie z regułą D2 można napisać

$$(a, v_1) + (a, v_2) = 1$$
$$(b, u_1) + (b, u_2) = 1$$
$$(c, w_1) + (c, w_2) = 1$$

Na podstawie reguły B9 można term t_4 przedstawić w postaci

$$t_5 = [(a, v_2) + (b, u_1)] \cdot [(a, v_1) + (a, v_2)] \cdot [(b, u_1) + (b, u_2)] \cdot [(c, w_1) + (c, w_2)]$$

skąd zgodnie z regułami A3, B3, B5, D1 otrzymamy term

$$\begin{aligned} t_6 = {} & (a, v_1) \cdot (b, u_1) \cdot (c, w_1) + \\ & + (a, v_1) \cdot (b, u_1) \cdot (c, w_2) + \\ & + (a, v_1) \cdot (b, u_2) \cdot (c, w_1) + \\ & + (a, v_1) \cdot (b, u_2) \cdot (c, w_2) + \\ & + (a, v_2) \cdot (b, u_1) \cdot (c, w_1) + \\ & + (a, v_2) \cdot (b, u_1) \cdot (c, w_2) + \\ & + (a, v_2) \cdot (b, u_2) \cdot (c, w_1) + \\ & + (a, v_2) \cdot (b, u_2) \cdot (c, w_2) \end{aligned}$$

który jest już w postaci normalnej. ■

Zauważmy, że term t_6 zawiera wszystkie możliwe termy proste w naszym języku, a suma wszystkich termów prostych jest równa 1.

Sprawdźmy to na przykładzie systemu określonego w przykładzie 2.3. W systemie tym istnieją jedynie trzy zbiory elementarne, którym odpowiadają termy proste

$$\sigma_S\, [(a, v_1) \cdot (b, u_1) \cdot (c, w_2)] = \{x_1, x_3\}$$
$$\sigma_S\, [(a, v_2) \cdot (b, u_1) \cdot (c, w_1)] = \{x_2, x_5, x_6\}$$
$$\sigma_S\, [(a, v_1) \cdot (b, u_2) \cdot (c, w_2)] = \{x_4\}$$

natomiast wszystkie pozostałe termy proste są puste, tj. odpowiadają im zbiory puste. A więc $\sigma_S(t_1) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, tzn. $t_1 = 1$.

Podany tu sposób sprowadzania termu do postaci normalnej jest niewygodny, gdyż wymaga on obliczania również wszystkich pustych termów prostych.

W konkretnym systemie informacyjnym S — metodę sprowadzania do postaci normalnej można znacznie uprościć, generując tylko niepuste termy proste przez pomnożenie termu sprowadzonego do postaci normalnej, nie przez sumę wszystkich możliwych termów prostych, lecz przez sumę niepustych w danym systemie termów prostych, tj. przez term charakterystyczny systemu t_S.

Przykład 2.5

Term

$$\sim(a, v_2)+(b, u_2)\cdot(c, w_2)$$

w systemie rozważanym w przykładzie 2.3 można sprowadzić do postaci normalnej w następujący sposób:

$$\begin{aligned}
&\sim(a, v_2)+(b, u_2)\cdot(c, w_2)=(a, v_1)+(b, u_2)\cdot(c, w_2)=\\
&=[(a, v_1)+(b, u_2)\cdot(c, w_2)]\ \{[(a, v_1)\cdot(b, u_1)\cdot(c, w_2)]+\\
&+[(a, v_2)\cdot(b, u_1)\cdot(c, w_1)]+[(a, v_1)\cdot(b, u_2)\cdot(c, w_2)]\}=\\
&=(a, v_1)\cdot(b, u_1)\cdot(c, w_2)+(a, v_1)\cdot(b, u_2)\cdot(c, w_2)
\end{aligned}$$

Ponieważ powyższym termom prostym odpowiadają w naszym systemie zbiory elementarne $\{x_1, x_3\}$ oraz $\{x_4\}$, więc odpowiedzią na pytanie

$$\sim(a, v_2)+(b, u_2)\cdot(c, w_2)$$

w omawianym systemie jest zbiór

$$\{x_1, x_3, x_4\}$$

tj.

$$\sigma_S[\sim(a, v_2)+(b, u_2)\cdot(c, w_2)]=\{x_1, x_3, x_4\}$$ ■

Definicja

Postać normalną, w której nie występują termy puste, będziemy nazywać *minimalną postacią normalną ze względu na system S.* ■

Zauważmy, że podany algorytm sprowadzania do postaci normalnej polega na usunięciu najpierw z termu negacji, a następnie sprawdzeniu, czy kolejne składniki termu są zawarte w niepustych termach prostych. Termy proste, dla których to zawieranie zachodzi, wchodzą w skład postaci normalnej. Opis ten nie jest zbyt ścisły, ale daje wskazówkę, jak zbudować prosty algorytm sprowadzania do postaci normalnej w konkretnym systemie.

Korzystając z możliwości przedstawienia termu w postaci normalnej, można bardzo prosto znaleźć odpowiedź na dowolne pytanie w systemie. Postać normalna mówi, że odpowiedź na dowolne pytanie jest zawsze sumą teoriomnogościową pewnych zbiorów elementarnych. Mając więc pytanie sprowadzone do postaci normalnej, możemy łatwo znaleźć odpowiadające każdemu termowi prostemu zbiory elementarne i w ten sposób otrzymać odpowiedź na zadane pytanie.

Można oczywiście odpowiedź na dowolne pytanie otrzymać także bez sprowadzania go do postaci normalnej, obliczając — zgodnie z definicją semantyki — wartości poszczególnych elementów pytania. Metoda taka jednak jest dość niewygodna, szczególnie w przypadku komputerowej realizacji systemu informacyjnego. Sprawę tę dokładniej omówimy w dodatku dotyczącym realizacji systemów informacyjnych.

Zauważmy na zakończenie tego punktu, że jeżeli system informacyjny nie jest selektywny, to w języku systemu nie można opisać dowolnego podzbioru zbioru obiektów systemu, a tylko te podzbiory które są sumami teoriomnogościowymi zbiorów elementarnych systemu. Zbiory takie będziemy nazywać *zbiorami opisywalnymi w systemie*. Ściślej:

Definicja

Niech $S = \langle X, A, V, \varrho \rangle$. Zbiór $Y \subset X$ nazwiemy zbiorem opisywalnym w systemie S wtedy i tylko wtedy, gdy istnieje taki term $t \in L_S$, że $\sigma_S(t) = Y$. ■

Zbiory opisywalne w S są to więc jedyne zbiory, które mogą stanowić odpowiedź na pytania zadawane w języku systemu. A więc nie każdy podzbiór zbioru obiektów może być opisany za pomocą odpowiedniego termu w języku systemu. Warto zdać sobie dokładnie sprawę z tego faktu, który jakkolwiek jest oczywisty, często nie jest dostrzegalny.

2.6. DOKŁADNOŚĆ I EFEKTYWNOŚĆ JĘZYKA

Pojęcie zbioru opisywalnego prowadzi do pytania, jak dokładnie można opisywać w języku informacyjnym zbiór obiektów systemu. Pojęcie dokładności nie jest tu ściśle sprecyzowane. Intuicyjnie chodzi o to, że język dokładniejszy pozwala otrzymać więcej odpowiedzi niż język mniej dokładny. Wiąże się to z faktem, że im więcej mamy zbiorów elementarnych w systemie, tym więcej otrzymamy zbiorów opisywalnych w tym systemie. W konsekwencji możliwych jest więcej odpowiedzi w systemie dokładniejszym niż w systemie mniej dokładnym. Ściśle biorąc, tak rozumiane pojęcie dokładności wiąże się raczej z pojęciem systemu informacyjnego niż z językiem zapytań. Ponieważ jednakże z każdym językiem wiążemy jakiś system informacyjny, przyjmiemy, że dokładność języka jest określona przez dokładność związanego z nim systemu informacyjnego. *Dokładność* tę określimy jako

$$\lambda_S = \frac{2^k}{2^{\operatorname{card}(X)}} = 2^{k - \operatorname{card}(X)}$$

gdzie k jest liczbą bloków elementarnych w systemie, X zaś zbiorem wszystkich obiektów w systemie informacyjnym S.

Współczynnik dokładności λ_S (lub λ, gdy system jest ustalony) jest więc określany przez stosunek liczby wszystkich podzbiorów opisywalnych w systemie S do liczby wszystkich możliwych podzbiorów zbioru obiektów w systemie S.

Takie pojęcie dokładności jest zgodne z intuicją i pozwala na liczbową ocenę dokładności języka zapytań. Przykładowo język systemu informacyjnego rozpatrywanego w przykładzie 2.3 ma dokładność

$$\lambda = \frac{2^3}{2^6} = 0{,}125$$

W systemie tym mamy trzy zbiory elementarne, można więc skonstruować $2^3 = 8$ zbiorów opisywalnych, tzn. możliwych jest w tym systemie tylko 8 różnych następujących odpowiedzi:

$\emptyset$ — zbiór pusty

$\{x_4\}$

$\{x_1, x_3\}$

$\{x_2, x_5, x_6\}$

$\{x_4\} \cup \{x_1, x_3\}$

$\{x_4\} \cup \{x_2, x_5, x_6\}$

$\{x_1, x_3\} \cup \{x_2, x_5, x_6\}$

$\{x_4\} \cup \{x_1, x_3\} \cup \{x_2, x_5, x_6\}$

Natomiast wszystkich podzbiorów zbioru obiektów systemu jest $2^6 = 64$.

Oczywiście najdokładniejszy jest każdy system selektywny i jego dokładność $\lambda = 1$. Każdy inny system ma dokładność mniejszą od 1.

Oprócz dokładności języka będzie nas interesować również dokładność konkretnej odpowiedzi w języku. Zdefiniujemy ją za pomocą pojęć wprowadzonych do tej pory.

Niech $S = \langle X, A, V, \varrho\rangle$ będzie systemem informacyjnym i niech $Y \subseteq X$ będzie dowolnym podzbiorem X.

Definicja

Y' jest *górnym przybliżeniem* Y w systemie S, gdy jest najmniejszym zbiorem opisywalnym w S zawierającym Y. ■

Definicja

Y'' jest *dolnym przybliżeniem* Y w systemie S, gdy jest największym zbiorem opisywalnym w S zawartym w Y. ■

Oczywiście górne i dolne przybliżenie każdego podzbioru obiektów jest dokładnie jedno. W przypadku systemu selektywnego dla każdego Y oba przybliżenia Y są sobie równe oraz są równe zbiorowi Y. Na przykład przybliżeniem górnym zbioru $\{x_1, x_4\}$ w systemie rozważanym w p. 2.5 jest zbiór $\{x_4\} \cup \{x_1, x_3\}$, a przybliżeniem dolnym tego zbioru jest zbiór $\{x_4\}$.

Definicja

Dokładnością górnego przybliżenia zbioru Y w systemie S nazwiemy stosunek

$$\frac{\operatorname{card}(\bar{S}Y)}{\operatorname{card}(Y)}$$

a *dokładnością dolnego przybliżenia zbioru Y* w systemie S nazwiemy stosunek

$$\frac{\operatorname{card}(\underline{S}Y)}{\operatorname{card}(Y)}$$

gdzie $\underline{S}Y$, $\bar{S}Y$ oznaczają odpowiednio dolne i górne przybliżenie zbioru Y w systemie S. ■

W języku systemu informacyjnego S nie można w ogólnym przypadku opisać dowolnego podzbioru Y zbioru obiektów X w systemie S, można jedynie opisać górne i dolne przybliżenie podzbioru Y. Gdybyśmy więc oczekiwali, na przykład na podstawie doświadczenia, że odpowiedzią na pytanie powinien być zbiór Y, a zbiór ten nie jest opisywalny w rozpatrywanym systemie, jako odpowiedź moglibyśmy otrzymać jedynie jego przybliżenie górne bądź dolne. Stąd właśnie koncepcja odpowiedzi przybliżonej. Wprowadźmy jeszcze jedno pojęcie charakteryzujące język zapytań systemu, mianowicie pojęcie efektywności języka systemu. Będzie ono charakteryzowało, jaka „część" języka może być faktycznie używana do zadawania pytań. W ogólnym przypadku bowiem wiele termów prostych języka w danym systemie jest pustych, i są one w pewnym sensie „bezużyteczne".

Definicja

Efektywność języka w systemie $S = \langle X, A, V, \varrho \rangle$ jest określona wzorem

$$\eta_S = \frac{k}{\prod_{a \in A} \operatorname{card}(V_a)}$$

gdzie k jest liczbą zbiorów elementarnych w systemie S. ■

W mianowniku tego wyrażenia znajduje się liczba wszystkich termów prostych w systemie S, w liczniku zaś liczba termów prostych niepustych. (Jest ich oczywiście tyle, ile jest bloków elementarnych w systemie).

Oczywiście $0 < \eta_S \leqslant 1$. Wartość 1 współczynnik efektywności przyjmuje dla systemu zupełnego. Dla każdego innego systemu jest on mniejszy od jedności. Tak więc współczynnik η_S określa, jak duża część języka systemu ma niepuste znaczenie w S.

Tak więc każdy system selektywny ma największą dokładność, a każdy system zupełny — największą efektywność.

2.7. INSTRUKCJE

W rozważanym do tej pory języku odpowiedzią na pytanie był podzbiór obiektów systemu. W istocie jesteśmy zainteresowani nie tyle uzyskaniem odpowiedzi w postaci zbioru obiektów, lecz raczej — dokumentów (informacji) odpowiadających obiektom stanowiącym odpowiedź. Tak rozumiana odpowiedź jest zbiorem wierszy tablicy wyznaczonych przez term (pytanie) t, tj.

$$\{\varrho_x\}_{x \in \sigma_S(t)}$$

Mówiąc jeszcze inaczej, odpowiedzią na pytanie t w systemie S jest podsystem S' systemu S taki, że

$$S' = S|_{\sigma_S(t)}$$

a więc podsystem systemu S z ograniczonymi obiektami.

Dla uniknięcia nieporozumień będziemy w dalszym ciągu ten drugi sposób odpowiedzi oznaczać przez

$$\sigma_S^*(t) = S|_{\sigma_S(t)}$$

Często jesteśmy zainteresowani nie tylko uzyskaniem zbioru obiektów (dokumentów) spełniających pewne warunki wyrażone w pytaniu, lecz chcemy także znać na przykład wartości pewnych atrybutów w wyszukanym zbiorze bądź też wykonać pewne operacje na wartościach określonych atrybutów. Możemy być zainteresowani np. średnim wynagrodzeniem pracowników określonego zakładu lub zawodu, bądź też zarobkami największymi lub najmniejszymi w danej grupie. Możemy być też zainteresowani tym, kto zarabia najmniej bądź najwięcej w danej grupie zawodowej. Oprócz wykonywania operacji na wartościach atrybutów mogą być też potrzebne informacje o relacjach zachodzących między wartościami określonych atrybutów. Na przykład, jeżeli chcemy znaleźć zbiór osób, które rozpoczęły pracę przed ukończeniem studiów, musimy umieć sprawdzić, czy data rozpoczęcia pracy jest wcześniejsza od daty ukończenia studiów — a więc sprawdzić zachodzenie pewnej relacji (mniejszości) między wartościami atrybutu DATA ROZPOCZĘCIA PRACY oraz DATA UKOŃCZENIA STUDIÓW. Podobnych przykładów, w których samo wyszukanie odpowiednich dokumentów nie jest wystarczające, można podać wiele. Rozważany do tej pory język nie pozwala na formułowanie pytań tego rodzaju. Aby zadawanie tego rodzaju pytań było możliwe, należy odpowiednio rozszerzyć język zapytań oraz zdefiniować właściwie jego semantykę. Można by to zrobić w podobny sposób, jak to zrobiliśmy dla języka termów rozpatrywanego do tej pory. Wymagałoby to jednak wprowadzenia wielu nowych formalnych definicji i wydłużyłoby znacznie tok naszego rozumowania. Ponieważ zagadnienia te nie stanowią głównego tematu naszych rozważań, problem ten omówimy pobieżnie, posługując się przykładami. W tym celu wprowadzimy i omówimy pojęcie instrukcji. Podamy kilka najważniejszych typów instrukcji pozwalających na wykonywanie pewnych operacji na wartościach atrybutów — a w konsekwencji rozszerzających język zapytań, tak aby można w nim było formułować pytania takie, jak to pokazano powyżej. Mimo że nie podamy ścisłych definicji rozważanych instrukcji, dla Czytelnika z pewnym doświadczeniem praktycznym w programowaniu powinny one być zrozumiałe.

2.8. INSTRUKCJE LICZBOWE

Często zanim uzyskamy odpowiedź na pytanie w postaci zbioru dokumentów, jesteśmy zainteresowani uzyskaniem najpierw informacji o liczbie obiektów (dokumentów) w odpowiedzi. W celu uzyskiwania odpowiedzi na tego rodzaju pytania wprowadzimy pojęcie instrukcji liczbowej.

Definicja

Instrukcja liczbowa jest funkcją

$$\mathrm{CARD}_S: L_S \rightarrow N$$

gdzie: L_S — zbiór termów systemu S; N — zbiór liczb naturalnych. ■

Wartość tej funkcji będziemy nazywać *kardynalnością odpowiedzi* i określimy, jak następuje:

$$\mathrm{CARD}_S(t) = \mathrm{Card}(\sigma_S(t)) = \sum_{i=1}^{n} \mathrm{Card}(\sigma_S(t_i))$$

przy czym $t = t_1 + t_2 + \ldots + t_k$ jest postacią normalną termu t. ■

Tak więc, znając liczbę elementów każdego zbioru elementarnego, można z łatwością obliczyć kardynalność każdego termu przez sprowadzenie go najpierw do postaci normalnej, a następnie zsumowanie kardynalności wszystkich termów prostych wchodzących w skład postaci normalnej rozważanego termu.

Liczba elementów każdego zbioru elementarnego może być policzona wcześniej i odpowiednio zmagazynowana w pamięci maszyny tak, aby ostatecznie obliczanie liczności odpowiedzi mogło być znacznie przyspieszone, gdyż wystarczy wtedy tylko dodać k liczb, zamiast sumować liczbę wszystkich obiektów stanowiących odpowiedź. Ma to szczególne znaczenie, gdy odpowiedź zawiera bardzo dużą liczbę obiektów (dokumentów), np. 10 000, natomiast liczba zbiorów elementarnych wchodzących w skład odpowiedzi może być niewielka, np. 100.

Ważniejszą sprawą jest tu zresztą nie sam czas sumowania, lecz konieczność — w przypadku sumowania wszystkich obiektów odpowiedzi — czytania ich z pamięci wolnej (np. dyskowej), co wymaga dużych czasów (wolny dostęp do pamięci tego rodzaju). W drugim przypadku natomiast nie jest konieczne odczytywanie pamięci wolnej, gdyż wszystkie dane o zbiorach elementarnych mogą być umieszczone w szybkiej pamięci operacyjnej.

Czytelnika zainteresowanego szerszym omówieniem tego tematu odsyłamy do pracy Lipskiego i Marka [53].

2.9. INSTRUKCJE NUMERYCZNE

Następną grupę stanowią pytania, na które odpowiedź jest również liczbą, jednakże nie jest to liczba elementów jakiegoś zbioru, lecz liczba otrzymana w wyniku rachunku numerycznego na wartościach atrybutów, zakładając oczywiście, że wartości atrybutów, o których tu mowa, należą do zbioru liczb rzeczywistych. Konieczność tego typu obliczeń występuje przede wszystkim w przypadku pytań o charakterze statystycznym, gdy pytamy o średnie, wielkości minimalne, maksymalne itd.

Instrukcje służące do tego celu nazwiemy *instrukcjami numerycznymi*. Zauważmy, że obliczanie instrukcji numerycznych składa się z trzech kroków: w pierwszym znajdujemy zbiór dokumentów spełniających odpowiedni warunek opisany przez term t, a następnie odczytujemy z wyszukanych dokumentów wartości odpowiednich atrybutów i wykonujemy interesujące nas obliczenia.

Na przykład, jeżeli jesteśmy zainteresowani średnią wieku ludzi zarabiających np. 6000 zł — musimy najpierw wyszukać wszystkie osoby (ich dokumenty) zarabiające 6000 zł miesięcznie, a następnie zsumować wartości atrybutu WIEK i obliczyć średnią.

Zauważmy również, że rozpatrywane dalej obliczenia, mogą być uprzednio wykonane dla wszystkich zbiorów elementarnych w systemie, tak że ostateczne obliczenie sprowadza się do wykonania już znacznie prostszych obliczeń na danych przygotowanych wcześniej dla zbiorów elementarnych. Przedstawiona tu idea została zaproponowana w pracach [4] i [7].

W tym punkcie podamy kilka przykładów instrukcji numerycznych. Znaczenie tych instrukcji jest oczywiste i nie wymaga obszerniejszego komentarza.

Przyjmiemy, że w rozpatrywanych dalej przykładach term t ma postać normalną $t_1+t_2+\ldots+t_k$. Wprowadzimy ponadto oznaczenie

$$V_{a,t} = \{v \in V_a \colon \bigvee_{x\in\sigma_S(t)} \varrho_x(a) = v\}$$

Zbiór $V_{a,t}$ jest więc podzbiorem zbioru V_a ($V_{a,t} \subset V_a$); jest to zbiór wszystkich wartości atrybutu a, które mogą przyjmować obiekty należące do zbioru wyznaczonego przez term t.

Zdefiniujemy najpierw sumę wartości atrybutu a w systemie S:

$$\mathrm{SUM}_S(t, a) = \sum_{v\in V_{a,t}} v = \sum_{i=1}^{k} \sum_{v\in V_{a,t_i}} v$$

Najmniejszą wartość atrybutu a w systemie S wyrazimy jako

$$\mathrm{MIN}_S(t, a) = \mathrm{Min}\{V_{a,t}\} = \mathop{\mathrm{Min}}_{1\leqslant i\leqslant k} \mathrm{Min}\{V_{a,t_i}\}$$

Podobnie przedstawimy *wartość maksymalną atrybutu a* w systemie S:

$$\mathrm{MAX}_S(t, a) = \mathrm{Max}\{V_{a,t}\} = \mathop{\mathrm{Max}}_{1\leqslant i\leqslant k} \mathrm{Max}\{V_{a,t_i}\}$$

Na podstawie wprowadzonych definicji można określić wartość średnią atrybutu a w systemie S w następujący sposób:

$$\mathrm{MEAN}_S(t, a) = \frac{\mathrm{SUM}_S(t, a)}{\mathrm{CARD}_S(t)} = \frac{\sum\limits_{i=1}^{k} \mathrm{MEAN}_S(t_i, a)}{k}$$

wartość środkową zaś określimy jako

$$\mathrm{MED}_S(t, a) = \frac{\mathrm{MAX}_S(t, a) - \mathrm{MIN}_S(t, a)}{2}$$

Podobnie możemy określić instrukcje „działające" na wartościach dwu i więcej atrybutów. Na przykład iloczyn skalarny wartości dwu atrybutów a, b w systemie S można określić następująco:

$$\mathrm{SKAL}_S(t, a, b) = \sum_{x \in \sigma_S(t)} \varrho_x(a) \cdot \varrho_x(b) = \sum_{i=1}^{k} \sum_{x \in \sigma_S(t_i)} \varrho_x(a) \cdot \varrho_x(b) =$$

$$= \sum_{i=1}^{k} \mathrm{SKAL}_S(t_i, a, b)$$

Przypominamy raz jeszcze, że wartościami wszystkich powyższych instrukcji są liczby.

Wskaźnik S przy funkcji będziemy pomijać, gdy wiadomo, o jaki system chodzi.

Podamy teraz kilka przykładów.

Przykład 2.6

Załóżmy, że system informacyjny dotyczy magazynu, w którym magazynuje się jakieś części, i że w kartotekach tych części znajdują się między innymi informacje o dacie zakupu każdej części, ich cenie i firmie, w której część została zakupiona, materiale itd. Mamy więc w naszym systemie atrybuty

NAZWA CZĘŚCI
KOD CZĘŚCI
DATA ZAKUPU
CENA
DOSTAWCA
MATERIAŁ

Możemy być np. zainteresowani uzyskaniem odpowiedzi, ile kosztowały nas narzędzia określonego typu, kupione w lipcu 1980 r. u dostawców A, B, C i wykonane ze stali o kodzie K_1. Odpowiedzią na nie będzie właśnie suma zapłacona za interesujące nas części, tj. wartość instrukcji

$\mathrm{SUM}_S(t,$ CENA)

gdzie

$t =$ (NARZĘDZIE = wiertło) · (MATERIAŁ = stal K_1) · (DOSTAWCA =
= A, B, C) · (DATA ZAKUPU = lipiec 1980). ■

Przykład 2.7

Rozważmy inny system informacyjny, w którym mamy następujące atrybuty:

ZAWÓD
WIEK
STAŻ
PŁEĆ
WYNAGRODZENIE

Możemy być np. zainteresowani uzyskaniem informacji o największym (bądź najmniejszym) wynagrodzeniu kobiet w wieku poniżej 25 lat, pracujących 3 lata w zawodzie księgowych.

Term określający odpowiedni zbiór będzie miał następującą postać:

$t =$ (ZAWÓD = księgowa) · (WIEK $\leqslant$ 25) · (STAŻ = 3) · (PŁEĆ = k)

Odpowiednia instrukcja numeryczna będzie miała postać

MAX (t, WYNAGRODZENIE)

Jeżeli przyjmiemy następujący term

$s =$ (ZAWÓD = księgowa) · (STAŻ = 5) · (WYNAGRODZENIE = 5000) ·
· (PŁEĆ = k)

to instrukcja

MAX (s, WIEK)

będzie stanowiła następujące pytanie: Jaki jest maksymalny wiek kobiety pracującej w zawodzie księgowej ze stażem pięcioletnim? ■

Przykład 2.8

Rozważmy prosty system informacyjny na uczelni, ewidencjonujący studentów i posiadający między innymi następujące atrybuty:

KIERUNEK
ROK STUDIÓW
PRZEDMIOT
OCENA

Jeżeli chcemy zapytać o średnią ocenę z fizyki na czwartym roku studiów kierunku chemii, to musimy najpierw napisać term

$t =$ (KIERUNEK = Chemia) · (ROK STUDIÓW = 4) · (PRZEDMIOT =
= fizyka)

a następnie utworzyć instrukcję

MEAN (t, OCENA) ■

Ostatni przykład zilustruje term opisujący obliczanie iloczynu skalarnego.

Przykład 2.9

Rozpatrzmy system informacyjny zawierający dane o materiałach budowlanych, mający między innymi następujące atrybuty:

NAZWA
DATA ZAKUPU
ILOŚĆ
CENA JEDNOSTKOWA
DOSTAWCA

Gdybyśmy chcieli się dowiedzieć, ile kosztował cement zakupiony po 1.XI.1980 z Cementowni w Wierzbicy, należałoby najpierw napisać term

$t =$ (NAZWA = cement) · (DATA ZAKUPU > 1.XI.1980) · (DOSTAWCA = = Wierzbica)

a następnie instrukcję

SKAL(t, ILOŚĆ, CENA JEDNOSTKOWA)

której wartość stanowi żądaną informację. ■

Czasem możemy być zainteresowani informacją nie tylko, jakie jest np. najniższe wynagrodzenie określonej grupy osób, lecz także tym, kto otrzymuje najniższe wynagrodzenie.

Rozważany do tej pory język nie pozwala na formułowanie pytań tego rodzaju — można go jednak łatwo rozszerzyć w ten sposób, aby zadawanie tego rodzaju pytań było możliwe. Wprowadzimy w tym celu nowe instrukcje, których znaczeniem będą zbiory obiektów systemu:

$$\mathrm{MAX!}_S(t, a) = \{x \in \sigma_S(t) \colon \varrho_x(a) = \mathrm{MAX}_S(t, a)\} = \sigma_S(t \cdot (a, \mathrm{MAX}_S(t, a)))$$

$$\mathrm{MIN!}_S(t, a) = \{x \in \sigma_S(t) \colon \varrho_x(a) = \mathrm{MIN}_S(t, a)\} = \sigma_S(t \cdot (a, \mathrm{MIN}_S(t, a)))$$

$$\mathrm{MED!}_S(t, a) = \{x \in \sigma_S(t) \colon \varrho_x(a) = \mathrm{MED}_S(t, a)\} = \sigma_S(t \cdot (a, \mathrm{MED}_S(t, a)))$$

$$\mathrm{MEAN!}_S(t, a) = \{x \in \sigma_S(t) \colon \varrho_x(a) = \mathrm{MEAN}_S(t, a)\} = \sigma_S(t \cdot (a, \mathrm{MEAN}_S(t, a)))$$

Przykłady wyjaśnią dokładniej znaczenie wprowadzonych instrukcji.

Rozpatrzmy term, rozważany w poprzednim przykładzie:

$t =$ (ZAWÓD = księgowa) · (WIEK $\leqslant$ 25) · (STAŻ = 3) · (PŁEĆ = k)

oraz instrukcję

MAX! (t, WYNAGRODZENIE)

Znaczeniem tej instrukcji jest zbiór wszystkich księgowych w wieku poniżej 25 lat, ze stażem 3-letnim, otrzymujących największe wynagrodzenie.

Podobnie rozważany poprzedni przykład obecnie będzie miał postać

$t =$ (KIERUNEK = Chemia) · (ROK STUDIÓW = 4) · (PRZEDMIOT =
= fizyka)

MEAN! (t, OCENA)

i będzie oznaczał zbiór wszystkich studentów Chemii czwartego roku mających z fizyki ocenę średnią dla tej grupy.

Widać, że tak rozszerzony język ma duże możliwości formułowania pytań, o dość skomplikowanej już budowie. Dodajmy, że podane w tym punkcie instrukcje można ze sobą łączyć, tworząc nowe termy, jednakże problemem tym nie będziemy zajmować się tutaj bliżej, pozostawiając przemyślenie go zainteresowanym Czytelnikom — tu podamy jedynie kilka prostych przykładów takich instrukcji

$\mathrm{MAX}_S(\mathrm{MEAN}_S(t, a), b)$

$\mathrm{MAX!}_S(\mathrm{MEAN!}_S(t, a), b)$

$\mathrm{MIN!}_S(\mathrm{MAX!}_S(t, a), b)$

Pytanie: ile lat ma najstarszy student wydziału matematyki mający średnią ocenę z fizyki, zapiszemy tak:

MAX [MEAN! ((WYDZIAŁ = mat), FIZYKA), WIEK]

Gdybyśmy chcieli znaleźć wszystkich najstarszych studentów wydziału matematyki mających średnie oceny z fizyki, użylibyśmy pytania

MAX! [MEAN! ((WYDZIAŁ = mat), FIZYKA), WIEK]

Podobnie można konstruować bardziej złożone przykłady.

2.10. INSTRUKCJE RELACYJNE

Często jesteśmy zainteresowani wyszukaniem obiektów mających tę własność, że między wartościami niektórych jego atrybutów zachodzą jakieś związki. Na przykład możemy być zainteresowani wyszukaniem osób, których miejsce zamieszkania i miejsce urodzenia są jednakowe, tzn. wszystkich takich x, dla których

ϱ_x(ADRES ZAM) $= \varrho_x$(ADRES UR)

lub pisząc krócej

ADRES ZAM = ADRES UR

W systemie informacyjnym zawierającym informacje o częściach zamiennych można o każdej części uzyskać informacje dotyczące liczby sztuk tej części w magazynie oraz informację o normatywnym minimalnym zapasie tych części, który to poziom nie może być przekroczony. Możemy wtedy być zainteresowani wyszukaniem tych wszystkich części, dla których stan magazynu jest „bliski" stanu minimalnego.

W systemie informacyjnym zawierającym informacje o dacie ukończenia szkoły i dacie rozpoczęcia pracy każdej osoby, możemy być zainteresowani uzyskaniem informacji o osobach, które rozpoczęły prace przed ukończeniem szkoły, szukamy wtedy wszystkich obiektów x w systemie, dla których

$$\sigma_x(\text{DATA UKOŃCZENIA SZKOŁY}) > \sigma_x(\text{DATA ROZPOCZĘCIA PRACY})$$

lub krócej

DATA UKOŃCZENIA SZKOŁY > DATA ROZPOCZĘCIA PRACY

W medycznym systemie informacyjnym można uzyskać dane o każdym pacjencie z dwu okresów, w chwili przyjścia do szpitala i w chwili opuszczania szpitala. Możemy być w takim przypadku zainteresowani, jak zmieniały się pewne parametry pacjenta w czasie leczenia (chodzi tu oczywiście o parametry ilościowe), np. waga lub ciśnienie krwi. Możemy wtedy np. być zainteresowani znalezieniem tych wszystkich pacjentów leczonych na określoną chorobę, u których waga w czasie leczenia zmalała lub odwrotnie.

Przykładów takich można podać wiele. Dla umożliwienia szukania odpowiedzi na tego rodzaju pytania wprowadzimy nowy rodzaj instrukcji zwanych *instrukcjami relacyjnymi* — mającymi postać następującą:

$$\mathrm{REL}_S(t, R(a, b))$$

gdzie R jest symbolem pewnej relacji (najczęściej $=$, $<$, $>$), zaś a, b są atrybutami oraz

$$R \subset V_a \times V_b$$

Uwaga. Identyfikujemy tu relację R z jej nazwą także oznaczoną przez R.

Znaczenie tej instrukcji jest określone, jak następuje:

$$\mathrm{REL}_S(t, R(a, b)) = \{x \in \sigma_S(t) \colon R(\varrho_x(a), \varrho_x(b))\}$$

Tak więc, jeżeli chcemy sprawdzić, czy np. zapas łożysk kulkowych określonego typu zbliża się do niezbędnego minimum, np. jest od niego o 10 sztuk większy, napiszemy

$t = (\text{WYRÓB} = \text{łożysko}) \cdot (\text{KOD} = \text{Ł.K121X})$
$\mathrm{REL}(t, \text{ILOŚĆ} \geqslant \text{ZAPAS} + 10)$

Atrybut ILOŚĆ wyraża aktualną liczbę danego wyrobu w magazynie, atrybut ZAPAS zaś oznacza minimalny normatywny zapas danej części, który powinien się zawsze znajdować w magazynie.

Instrukcje relacyjne, podobnie jak poprzednie instrukcje, mogą być również obliczone najpierw dla zbiorów elementarnych, a następnie na tej podstawie może być liczona wartość dla dowolnego termu. A więc

$$\mathrm{REL}_S(t, R(a, b)) = \{x \in \sigma_S(t): R(\varrho_x(a), \varrho_x(b))\} = \bigcup_{i=1}^{k} \mathrm{REL}_S(t_i, R(a, b))$$

gdzie $t_1+t_2+ \ldots +t_k$ jest postacią normalną termu t.

2.11. INSTRUKCJE LOGICZNE

Wprowadzone w punkcie 2.3 pojęcie prawdziwości termu pozwala również formułować za pomocą termów pytania nazwane uprzednio logicznymi, na które odpowiedź jest postaci TAK, NIE lub PRAWDA, FAŁSZ.

W tym celu wprowadzimy dwa rodzaje instrukcji logicznych

$$t \underset{S}{=} t' \quad \text{oraz} \quad t \underset{S}{\leqslant} t'$$

Przypominamy, że

$$t \underset{S}{=} t', \text{ gdy } \sigma_S(t \leftrightarrow t') = X, \text{ tj. } \sigma_S(t) = \sigma_S(t')$$

Podobnie $t \underset{S}{\leqslant} t'$, gdy $\sigma_S(t \to t') = X$, tj. $\sigma_S(t) \subset \sigma_S(t')$

Sprawdzanie czy $t \underset{S}{=} t'$, jest bardzo proste. Jeżeli termy t i t' sprowadzimy do postaci normalnej $t \underset{S}{=} t_1+t_2+ \ldots +t_k$ oraz $t' \underset{S}{=} t_1'+t_2'+ \ldots +t_l'$, to równość termów t, t' będzie zachodzić wtedy i tylko wtedy, gdy

$$t_1+t_2+ \ldots +t_k \underset{S}{=} t_1'+t_2'+ \ldots +t_l'$$

Ta ostatnia równość zachodzi tylko wtedy, gdy występujące tylko po lewej lub prawej stronie równości termy proste są puste.

Podobnie, sprawdzenie relacji $t \underset{S}{\leqslant} t'$ można sprowadzić do sprawdzenia, czy w wyrażeniu

$$t_1+t_2+ \ldots +t_k \underset{S}{\leqslant} t_1'+t_2'+ \ldots +t_l'$$

gdzie $t_1+t_2+ \ldots +t_k$ jest postacią normalną termu t, $t_1'+t_2'+ \ldots +t_l'$ jest postacią normalną termu t'

wszystkie termy proste występujące po lewej stronie znaku $\leqslant$ i nie występujące po prawej stronie są puste. Jeśli tak, to relacja $t \underset{S}{\leqslant} t'$ zachodzi, w przeciwnym przypadku nie zachodzi.

Podany tu sposób sprawdzania prawdziwości termów jest bardzo wygodny z praktycznego punktu widzenia, pozwala on bowiem na uzyskanie odpowiedzi na pytanie bez konieczności obliczania wartości termów t, t' i ich porównywania (co może być bardzo mało efektywne) — a jedynie drogą formalnych przekształceń termów, co jest znacznie bardziej wygodne i ekonomiczne. Nie potrzeba wtedy w przypadku komputerowej realizacji systemu sięgać do wolnych pamięci masowych, wystarczy korzystać jedynie z szybkiej pamięci operacyjnej.

Podamy teraz kilka przykładów prostych pytań logicznych. Jeżeli w systemie informacyjnym o studentach występują atrybuty DOCHÓD oraz STYPENDIUM, przy czym atrybut dochód jest rozumiany jako dochód miesięczny, przypadający na osobę w rodzinie studenta, to instrukcja

$$(\text{DOCHÓD} \leqslant 1500 \text{ zł}) = (\text{STYPENDIUM} = \text{otrzymuje})$$

oznacza pytanie:
„Czy stypendia otrzymują tylko studenci o dochodzie miesięcznym na osobę w rodzinie mniejszym niż 1500 zł?".

Natomiast instrukcja

$$(\text{DOCHÓD} \leqslant 1500 \text{ zł}) \leqslant (\text{STYPENDIUM} = \text{otrzymuje})$$

oznacza pytanie: „Czy każdy student o dochodzie na osobę w rodzinie mniejszym niż 1500 zł miesięcznie otrzymuje stypendium?".

Jeżeli nierówność ta jest prawdziwa, znaczy to, że każdy student o dochodzie miesięcznym mniejszym niż 1500 zł na osobę otrzymuje stypendium — ale być może również inni studenci otrzymują stypendium.

Powyższe instrukcje można obliczyć podobnie, jak poprzednio, wyszukując zbiór studentów o dochodzie w rodzinie mniejszym niż 1500 zł na osobę miesięcznie oraz zbiór studentów otrzymujących stypendium, a następnie porównując te zbiory. Możemy postąpić też inaczej, sprowadzając termy

$$(\text{DOCHÓD} \leqslant 1500 \text{ zł}) \quad \text{oraz} \quad (\text{STYPENDIUM} = \text{otrzymuje})$$

do postaci normalnej i sprawdzając, czy odpowiednie termy proste w nich występujące są puste.

W rozpatrywanym przykładzie nie można wykonać sprowadzenia do postaci normalnej interesujących nas termów, gdyż nie podaliśmy ani systemu informacyjnego, ani jego języka. Jednakże sprawa ta jest bardzo prosta (patrz p. 2.5) i nie będziemy jej tutaj dyskutować dokładniej.

Pojęcie instrukcji logicznej można w prosty sposób rozszerzyć, porównując wartości instrukcji numerycznych. Na przykład:

$$\text{MAX}_S(t, a) = \text{MAX}_S(t', b)$$

$$\text{MEAN}_S(t, a) = \text{MEAN}_S(t', b)$$

$$\text{MAX}_S(t, a) = \text{MIN}_S(t', b)$$

itd.

Gdybyśmy chcieli się na przykład dowiedzieć, czy średnia ocen z fizyki studentów wydziału matematyki jest taka sama, jak średnia ocen z matematyki studentów wydziału fizyki, to należy znaleźć odpowiedź na pytanie wyrażone następująco:

MEAN((WYDZIAŁ = mat), FIZYKA) = MEAN((WYDZIAŁ = fiz), MATEMATYKA)

Atrybuty FIZYKA i MATEMATYKA oznaczają odpowiednio oceny z fizyki i matematyki.

Możemy też rozważać instrukcje logiczne postaci:

MAX!(t, a) = MAX!(t', b) itd.

Przykładem takiej instrukcji może być pytanie: „Czy osoby mające najdłuższy staż pracy w danym zakładzie mają najwyższe wynagrodzenie". Pytanie takie będzie miało postać:

MAX!((ZAWÓD = księgowy), STAŻ) = MAX!((ZAWÓD = księgowy), WYNAGRODZENIE)

Pytanie: Czy maksymalne zarobki osób w wieku poniżej 25 lat są równe minimalnym zarobkom osób powyżej 25 lat? zapiszemy za pomocą instrukcji

MAX((WIEK < 25), WYNAGRODZENIE) = MIN((WIEK > 25), WYNAGRODZENIE)

Można podać ogólną postać instrukcji logicznych zawierających instrukcje numeryczne, jednak nie zrobimy tego, gdyż nie wszystkie takie instrukcje są interesujące z praktycznego punktu widzenia.

Na zakończenie tego punktu przypominamy raz jeszcze, że wartości wszystkich dyskutowanych tu termów można obliczać na podstawie odpowiednich wartości obliczanych dla zbiorów elementarnych.

2.12. INSTRUKCJE PORZĄDKUJĄCE

Instrukcje porządkujące służą do porządkowania zbiorów według jakiegoś przyjętego kryterium. Omówimy tu najprostszy przypadek tego rodzaju instrukcji, którą oznaczymy przez

$\mathrm{ORD}_S(t, a)$

Instrukcja ta powoduje uporządkowanie zbioru obiektów $\sigma_S(t)$ według wartości atrybutu a, zakładając oczywiście, że w zbiorze V_a jest określona relacja porządku. Na przykład, jeżeli zbiorem obiektów są pracownicy jakiejś instytucji, atrybutem a zaś jest nazwisko, to zbiór elementów można za pomocą tej instrukcji uporządkować leksykograficznie (według kolejności liter, tak jak słowa w słowniku). Jeżeli atrybutem *a* jest wiek, to instrukcja porządkująca porządkuje według wieku; w przypadku, gdy atrybutem *a* jest wynagrodzenie, obiekty zostaną uporządkowane przez tę instrukcję według wielkości wynagrodzenia.

Również w przypadku tej instrukcji można wykorzystać fakt podziału zbioru porządkowanych obiektów na zbiory elementarne, tzn. najpierw uporządkować elementy w zbiorach elementarnych, a następnie — standardowymi algorytmami — dokonać scalenia (ang. merging) uporządkowanych zbiorów elementarnych, uzyskując ostateczny rezultat.

3. SYSTEMY WIELOSTOPNIOWE I HIERARCHICZNE

3.1. WPROWADZENIE

W dotychczasowych rozważaniach dotyczących systemów informacyjnych i ich języków nie braliśmy pod uwagę liczności zbiorów obiektów, atrybutów i wartości. Łatwo zauważyć, że przy niedużej liczbie obiektów w systemie (np. kilkadziesiąt) oraz dużej liczbie atrybutów (np. kilkaset) i dużej liczbie ich wartości (np. kilka tysięcy) w języku systemu wystąpi bardzo wiele termów prostych, z tym, że znakomita ich większość będzie pusta, tj. odpowiadające im zbiory elementarne będą puste. Na przykład dla dziesięciu atrybutów, z których każdy przyjmuje 10 wartości, termów prostych jest 10^{10}, tj. 10000000000. Oczywiście w istniejących w rzeczywistości systemach liczba obiektów jest dużo mniejsza, a w konsekwencji również liczba zbiorów elementarnych jest nieporównywalnie mniejsza od liczby możliwych termów prostych. W rezultacie wykorzystanie własności systemów, które podaliśmy w poprzednich rozdziałach, byłoby praktycznie niemożliwe w rzeczywistości istniejących systemach informacyjnych. Dodajmy ponadto, że istotną rolę w wykorzystaniu podanych własności systemów informacyjnych odgrywa fakt, że różne atrybuty mogą się bardzo różnić liczbą możliwych wartości. Na przykład atrybut PŁEĆ może przyjmować tylko dwie wartości, atrybut NAZWISKO może natomiast przyjmować kilka czy nawet kilkadziesiąt tysięcy wartości. Powoduje to również, że zbiory elementarne systemu mogą mieć charakter bardzo niewygodny z punktu widzenia realizacji systemu.

Dla uniknięcia powyższych niedogodności musimy dokonać pewnych zabiegów, tak aby rozważane własności mogły znaleźć zastosowanie w rzeczywistych sytuacjach. Problemy te omówimy dokładniej w pozostałej części tego rozdziału.

3.2. SYSTEMY DWUSTOPNIOWE

Rozpatrzymy następującą sytuację:
Dany jest zredukowany system informacyjny $S=\langle X, A, V, \varrho\rangle$, tj. system, w którym wszystkie atrybuty są niezależne — oraz system $S'=\langle X, A', V, \varrho'\rangle$, który jest podsystemem systemu S z ograniczonymi atrybutami. Zachodzi wtedy, jak to podano w rozdz. 1, zależność

$$\underset{S'}{\sim} \subset \underset{S}{\sim}$$

przy czym zawieranie to jest istotne. A więc usuwanie atrybutów w zredukowanym systemie informacyjnym powoduje, że w tak otrzymanym systemie S' zbiory elementarne są „większe" i są sumami zbiorów elementarnych systemu wyjściowego S.

Jeżeli więc wyróżnimy w systemie $S = \langle X, A, V, \varrho\rangle$ pewien podzbiór atrybutów $B \subset A$ otrzymamy dwa podziały zbioru obiektów X, $\underset{S}{\sim} = \underset{A}{\sim}$ oraz $\underset{B}{\sim}$, z których drugi jest zawarty w pierwszym tj. $\underset{A}{\sim} \supset \underset{B}{\sim}$.

Zbiory elementarne generowane przez zbiór atrybutów A' będziemy nazywać *zbiorami elementarnymi pierwszego stopnia systemu S*, zbiory elementarne zaś generowane przez wyróżniony zbiór atrybutów B nazwiemy *zbiorami elementarnymi drugiego stopnia systemu S.*

System informacyjny z tak wyróżnionym zbiorem atrybutów będziemy nazywać *systemem dwustopniowym.* System taki będziemy oznaczać przez

$$S = \langle X, A, B, V, \varrho\rangle$$

Przykład 3.1

Rozważmy system informacyjny S określony tabelką podaną poniżej:

X	a	b	c
x_1	u_1	w_2	v_2
x_2	u_2	w_1	v_2
x_3	u_1	w_2	v_1
x_4	u_2	w_1	v_1
x_5	u_2	w_1	v_1
x_6	u_2	w_1	v_1
x_7	u_2	w_1	v_2
x_8	u_1	w_2	v_2

Zbiory elementarne pierwszego systemu S są następujące:

$$E_1 = \{x_3\}$$

$$E_2 = \{x_1, x_8\}$$

$$E_3 = \{x_2, x_7\}$$

$$E_4 = \{x_4, x_5, x_6\}$$

Jeżeli jako zbiór atrybutów wyróżnionych przyjmiemy $B = \{a, b\}$, to zbiorami elementarnymi drugiego stopnia systemu S będą:

$$E'_1 = \{x_1, x_3, x_8\}$$

$$E'_2 = \{x_2, x_4, x_5, x_6, x_7\}$$

lub pisząc inaczej

$$E'_1 = E_1 \cup E_2$$

$$E'_2 = E_3 \cup E_4$$

■

Rozważmy teraz, w jaki sposób można uzyskać odpowiedź na pytanie w systemie dwustopniowym.

W wyniku wprowadzenia wyróżnionego podzbioru atrybutów mamy w systemie dwustopniowym w istocie dwa języki zapytań: jeden związany ze zbiorem atrybutów A, drugi — ze zbiorem atrybutów B, które oznaczymy odpowiednio przez L_A i L_B.

Mamy teraz dwie możliwości szukania odpowiedzi na pytanie. W pierwszym przypadku możemy postępować tak, jak gdyby system informacyjny był jednostopniowy, tj. zadane pytanie sprowadzamy do postaci normalnej i znajdujemy odpowiedź jako sumę zbiorów elementarnych pierwszego stopnia. Możemy też postąpić inaczej. Jeżeli pytanie należy do języka L_B, tj. jeżeli występują w nim jedynie atrybuty należące do zbioru B, to pytanie to sprowadzamy do postaci normalnej w języku L_B i odpowiedź na nie będzie sumą zbiorów elementarnych drugiego stopnia. Jeżeli natomiast pytanie zawiera atrybuty należące do zbioru $A-B$, wtedy zostawiając w nim jedynie atrybuty ze zbioru B i sprowadzając tak zmodyfikowane pytanie do postaci normalnej w języku L_B, otrzymamy pytanie przybliżone, na które odpowiedź jest sumą zbiorów elementarnych drugiego stopnia. Z naszych poprzednich rozważań wynika, że jeżeli $t \in L_A$, zaś $t|_B \in L_B$ jest pytaniem otrzymanym z pytania t przez usunięcie z t wszystkich atrybutów nie należących do B, to

$$\sigma_S(t) \subset \sigma_S(t|_B)$$

Przyjmiemy tu, że jeżeli w pytaniu t nie występują atrybuty należące do B, to $t|_B = t$. Znaczy to, że odpowiedź $\sigma_S(t)$ na pytanie t jest zawarta w odpowiedzi $\sigma_S(t|_B)$ na pytanie przybliżone $t|_B$.

Dla uzyskania dokładnej odpowiedzi należy teraz w odpowiedzi przybliżonej, tj. w zbiorze obiektów $\sigma_S(t|_B)$ znaleźć odpowiedź na pytanie $t|_{A-B}$, tj. uwzględnić te atrybuty, które nie były uwzględnione w uzyskaniu odpowiedzi przybliżonej. Dokładniej, odpowiedź na pytanie $t \in L_A$ w systemie dwustopniowym $S = \langle X, A, B, V, \varrho\rangle$ jest określona następująco:

$$\sigma_S(t) = \sigma_{S|\sigma_S(t|_B)}(t|_{A-B})$$

natomiast jeżeli $t \in L_B$, to

$$\sigma_S(t) = \bigcup_{i=1}^{k} \sigma_S(t_i)$$

gdzie t ma postać normalną $t_1 + t_2 + \ldots + t_k$.

W ten sposób w systemie dwustopniowym odpowiedź na pytanie otrzymuje się w jednym albo w dwu etapach. Jeżeli pytanie zawiera tylko atrybuty ze zbioru B, odpowiedź otrzymuje się w jednym etapie jako sumę zbiorów elementarnych drugiego stopnia, jeżeli pytanie t zawiera atrybuty spoza zbioru B, odpowiedź jest wyszukiwana w dwu etapach: w pierwszym etapie otrzymuje się odpowiedź przybliżoną będącą sumą zbiorów elementarnych drugiego stopnia, a następnie w odpowiedzi przybliżonej wyszukuje się obiekty spełniające warunki wyrażone przez atrybuty, które nie były uwzględnione w pierwszym etapie wyszukiwania. Drugi etap wyszukiwania może być oczywiście realizowany dowolną metodą, np. metodą przeglądu zupełnego.

Tak więc dwustopniowy system informacyjny pozwala nam, dzięki ograniczeniu liczby atrybutów, według których odbywa się wyszukiwanie w pierwszym etapie, uzyskać odpowiednią liczbę i wielkość zbiorów elementarnych (drugiego stopnia).

Przykład 3.2

W systemie informacyjnym, w którym są określone następujące atrybuty:

NAZWISKO

DATA URODZENIA

MIEJSCE URODZENIA

ADRES ZAMIESZKANIA

IMIONA RODZICÓW

OCZY

WŁOSY

WZROST

PŁEĆ

STAN CYWILNY

WIEK

zamiast formułować zbiory elementarne pierwszego stopnia na podstawie wszystkich atrybutów, można utworzyć system dwustopniowy, przyjmując jako atrybuty wyróżnione np. KOLOR OCZU, KOLOR WŁOSÓW i PŁEĆ i na ich podstawie utworzyć zbiory elementarne (drugiego stopnia). Jeżeli teraz w pytaniu występują tylko atrybuty wyróżnione, to sprowadzając pytanie do postaci normalnej, otrzymamy odpowiedź jako sumę zbiorów elementarnych drugiego stopnia. Jeżeli w pytaniu występują atrybuty nie należące do zbioru atrybutów wyróżnionych, to uzyskanie odpowiedzi będzie przebiegało w dwu etapach: w pierwszym — uzyskanie odpowiedzi przybliżonej uwzględniającej tylko atrybuty wyróżnione i w drugim — w którym wyszukiwanie dokładnej odpowiedzi w zbiorze obiektów uzyskanym w pierwszym etapie (odpowiedzi przybliżonej) przebiega na podstawie pozostałych atrybutów pytania. Zakładamy tu, że wszystkie pytania występują w postaci normalnej.

Jeżeli więc nasze pytanie ma postać:

$$(\text{OCZY} = \text{niebieskie}) \cdot (\text{PŁEĆ} = \text{kobieta})$$

to odpowiedź na nie będzie sumą odpowiednich zbiorów elementarnych drugiego stopnia.

Natomiast dla znalezienia odpowiedzi na pytanie:

$$(\text{OCZY} = \text{niebieskie}) \cdot (\text{PŁEĆ} = \text{kobieta}) \cdot (\text{STAN CYWILNY} = \text{mężatka})$$

musielibyśmy najpierw uzyskać odpowiedź przybliżoną uwzględniającą atrybuty OCZY, WŁOSY, tj. uzyskać odpowiedź na pytanie

(OCZY = niebieskie) · (PŁEĆ = kobieta)

a następnie wśród wszystkich kobiet o oczach niebieskich, wyszukać wszystkie zamężne.

Gdyby nasze pytanie miało postać:

(MIEJSCE URODZENIA = Warszawa)

a więc nie występowałyby w nim atrybuty wyróżnione, wówczas pierwszy etap wyszukiwania odpowiedzi przybliżonej nie miałby sensu i należałoby dla uzyskania odpowiedzi przejść od razu do drugiego etapu, tj. do wyszukania wszystkich osób urodzonych w Warszawie — w całym zbiorze obiektów systemu. Ściślej biorąc, można w takim przypadku uważać, że cały zbiór obiektów jest tu przybliżoną odpowiedzią, tak więc również i w tym przypadku, z formalnego punktu widzenia można uznać, że proces wyszukiwania odbywa się w dwu etapach.

3.3. SYSTEMY WIELOSTOPNIOWE

Podana w poprzednim punkcie metoda może być uogólniona na przypadek więcej niż dwuetapowego wyszukiwania (na ogół w praktyce wystarczające są trzy etapy wyszukiwania).

Definicja

Niech $S = \langle X, A, V, \varrho \rangle$ będzie zredukowanym systemem informacyjnym i niech będzie dany ciąg $A_1, A_2, \ldots, A_n$, gdzie $A_1 = A, A_{i+1} \subset A_i$ dla $0 \leqslant i < n$. Wtedy $\underset{A_i}{\sim} \subset \underset{A_{i+1}}{\sim}$ dla każdego i $(0 \leqslant i < n)$, będą ustalonymi podzbiorami atrybutów zbioru A.

System informacyjny $S = \langle X, A_1, A_2, \ldots, A_n, V, \varrho \rangle$ będziemy nazywali *n-stopniowym systemem informacyjnym.* ■

Algorytm szukania odpowiedzi na dowolne pytanie t w takim systemie polega najpierw na zakwalifikowaniu pytania t do odpowiedniego języka L_{A_i}, a następnie postępowaniu w podobny sposób, jak to określimy dla systemów dwustopniowych.

Zauważmy, że istnieje tu jednak pewna niejednoznaczność postępowania. Jeżeli pytanie t zawiera atrybuty należące do zbioru $A_i - A_{i+1}$, to można w takim przypadku postępować dwojako. Po pierwsze można sprowadzić pytanie t do postaci normalnej w języku L_{A_i} i otrzymać odpowiedź jako sumę zbiorów elementarnych relacji $\underset{A_i}{\sim}$, a więc otrzymać odpowiedź w jednym kroku. Można też postąpić inaczej, a mianowicie znaleźć najpierw odpowiedź na pytanie $t|_{A_{i+1}}$ jako sumę zbiorów elementarnych relacji $\underset{A_{i+1}}{\sim}$, która jest oczywiście odpowiedzią przybliżoną, a następnie w tej odpowiedzi znaleźć obiekty według pozostałych atrybutów, tj. znaleźć odpowiedź na pytanie $t|_{A_i - A_{i+1}}$. To, który z możliwych dwu sposobów szukania odpowiedzi jest bardziej efektywny, zależy od wielu czynników i wymaga szczegółowej analizy. Problem ten wykracza jednak poza ramy niniejszej książki.

Na zakończenie zilustrujemy podane rozważania przykładem. Rozpatrzmy system informacyjny z atrybutami, takimi jak w przykładzie 3.2.

Przykład 3.3

Jako A_1 przyjmiemy zbiór wszystkich atrybutów systemu. Pozostałe zbiory atrybutów określimy następująco:

$$A_2 = \{\text{WIEK, PŁEĆ, ADRES ZAMIESZKANIA}\}$$

$$A_3 = \{\text{WIEK, PŁEĆ}\}$$

$$A_4 = \{\text{PŁEĆ}\}$$

Jeżeli w zadanym pytaniu występują tylko atrybuty WIEK, PŁEĆ, ADRES ZAMIESZKANIA, to po sprowadzeniu go do postaci normalnej w języku L_{A_2} można otrzymać odpowiedź bezpośrednio jako sumę zbiorów elementarnych relacji A_2. Można też uzyskać odpowiedź w dwu etapach, najpierw znaleźć odpowiedź przybliżoną, uwzględniając tylko atrybuty WIEK i PŁEĆ, a następnie w tak otrzymanym zbiorze wyszukać odpowiedź dokładną, uwzględniając atrybut ADRES. ■

3.4. SYSTEMY HIERARCHICZNE

Liczbę zbiorów elementarnych w systemie informacyjnym można również zmniejszyć w inny sposób niż to podano w poprzednich punktach, a mianowicie przez grupowanie wartości atrybutów. Rozpatrzmy najpierw prosty przykład ilustrujący ten problem.

Atrybut o dużej liczbie wartości powoduje generowanie dużej liczby zbiorów elementarnych w systemie. Na przykład atrybut NAZWISKO może mieć bardzo dużą liczbę wartości. Jeżeli system informacyjny dotyczy np. ewidencji ludności, to może w nim występować kilkadziesiąt czy kilkaset tysięcy nazwisk. W konsekwencji gdybyśmy uwzględnili wszystkie te nazwiska przy tworzeniu termów prostych, a w rezultacie również zbiorów elementarnych, otrzymalibyśmy zbyt dużą liczbę zbiorów elementarnych w systemie. Możemy jednak nazwiska pogrupować według np. pierwszej litery i utworzyć odpowiednie zbiory elementarne nie według całego nazwiska, lecz według pierwszej litery nazwiska. W ten sposób szukając odpowiedzi np. na pytanie:

$$(\text{NAZWISKO} = \text{Kowalski}) \cdot (\text{WIEK} < 25)$$

znajdziemy najpierw wszystkie osoby, których nazwiska zaczynają się na literę K mające poniżej 25 lat, a następnie wśród tych osób możemy już wyszukać osoby o nazwisku Kowalski.

W tym przypadku konieczne jest, jak w poprzedniej metodzie, wyszukiwanie dwustopniowe. W pierwszym etapie na podstawie pierwszej litery nazwiska otrzymamy, podobnie jak poprzednio, odpowiedź przybliżoną, a następnie w drugim etapie, uwzględniając całe nazwiska, otrzymamy odpowiedź dokładną.

Podobna sytuacja zachodzi np. dla atrybutu WIEK. Wartościami tego atrybutu mogą być liczby naturalne (gdy wiek określamy z dokładnością do jednego roku), ale możemy również wprowadzić wartości „młody", „średni", „stary". Wartość „młody" może oznaczać wiek poniżej 30 lat, „średni" — od 30 do 50 lat i „stary" — powyżej 50 lat.

Tak więc pytając o osoby w wieku młody, średni lub stary, można bezpośrednio uzyskać odpowiedź w jednym etapie, jeżeli natomiast chcielibyśmy znaleźć osoby w wieku między 25 a 40 lat lub osoby poniżej 35 lat, musielibyśmy zastosować tu również dwustopniowy etap wyszukiwania, wiedząc, że osoby w wieku poniżej 35 lat należy szukać w zbiorach osób młodych i w wieku średnim.

Podobnie sytuacja przedstawia się w przypadku atrybutów takich jak WZROST (niski, średni, wysoki), MOC SILNIKA (mała, średnia, duża,) WAGA itd.

Grupowanie wartości atrybutu polega w istocie na dodaniu nowego atrybutu zależnego od atrybutu, którego wartość grupujemy. Na przykład grupowanie nazwisk według pierwszej litery oznacza wprowadzenie nowego atrybutu do systemu, a mianowicie atrybutu PIERWSZA LITERA NAZWISKA. Oczywiście

NAZWISKO $\rightarrow$ PIERWSZA LITERA NAZWISKA

Podobnie można uważać, że podzielenie wartości atrybutu WIEK na grupy polega na wprowadzeniu nowego atrybutu KATEGORIA WIEKU, który oczywiście jest zależny od atrybutu WIEK, tj.

WIEK $\rightarrow$ KATEGORIA WIEKU

Podobnie sytuacja przedstawia się dla innych dyskutowanych tu przykładów.

Grupowanie takie może być również wielostopniowe, tzn. możemy najpierw pogrupować wartości atrybutu według jakiegoś kryterium, a następnie tak otrzymane grupy połączyć w większe grupy itd.

Metoda ta polega w istocie na tym, że jeżeli mamy niezredukowany system informacyjny $S = \langle X, A, V, \varrho \rangle$, to jako zbiory elementarne tego systemu bierzemy nie bloki relacji $\underset{S}{\sim} = \underset{A}{\sim}$, lecz bloki relacji $\underset{B}{\sim}$, gdzie $B \subset A$ oraz B nie zawiera żadnego takiego atrybutu $a \in A$, dla którego istnieje jej atrybut $b \in B$, taki że $a \rightarrow b$.

Systemy takie będziemy nazywać *systemami hierarchicznymi pierwszego stopnia* i będziemy je oznaczać:

$$S = \langle X, A, B, V, \varrho \rangle$$

Oczywiście system

$$S' = \langle X, B, V, \varrho' \rangle$$

gdzie

$$\varrho' = \varrho|_{X \times B}$$

jest systemem zredukowanym, oraz oczywiście $\underset{S}{\sim} \subset \underset{S'}{\sim}$.

Można również wprowadzić pojęcie systemu hierarchicznego wyższego stopnia, jednakże sprawą tą nie będziemy się tutaj zajmować.

Otrzymywanie odpowiedzi w systemie hierarchicznym może być jedno- albo dwuetapowe. Jeżeli pytanie jest zadane w języku L_A, to najpierw jest formułowane pytanie przybliżone w języku L_B i następnie w drugim etapie w odpowiedzi na pytanie przybliżone jest wyszukiwana pełna odpowiedź według oryginalnego pytania. Na przykład szukając odpowiedzi na pytanie, w którym występuje nazwisko Kowalski, najpierw szukamy wszystkich nazwisk rozpoczynających się na literę K, a następnie w tym zbiorze szukamy wszystkich osób o nazwisku Kowalski. Jeżeli zaś pytanie sformułowane jest w języku L_B, to odpowiedź otrzymujemy w jednym etapie. Na przykład jeżeli pytanie dotyczy wyszukania wszystkich osób młodych, to zbiór tych osób można otrzymać jako sumę zbiorów elementarnych relacji $\underset{B}{\sim}$.

3.5. UWAGI KOŃCOWE

Systemy wielostopniowe i hierarchiczne pozwalają na otrzymanie odpowiedniej wielkości zbiorów elementarnych w systemie informacyjnym, co jest konieczne z uwagi na efektywność systemu opartego na metodzie składowych atomowych. W istocie praktycznie niemal każdy system informacyjny realizowany w oparciu o tę metodę wymaga realizacji wielostopniowej bądź hierarchicznej — w przeciwnym przypadku bowiem realizacja taka byłaby całkowicie nieefektywna.

Powoduje to jednocześnie konieczność uzyskiwania odpowiedzi w niektórych przypadkach w dwu etapach: najpierw uzyskanie odpowiedzi przybliżonej, a następnie dopiero na jej podstawie — odpowiedzi dokładnej.

Szczegółowa analiza tego problemu wymagałaby jednak znacznie bardziej rozbudowanego aparatu formalnego oraz wielu założeń upraszczających, dlatego tą sprawą nie będziemy się tu bliżej zajmować.

Dodajmy jeszcze na zakończenie tego rozdziału, że inną koncepcję systemu hierarchicznego podali W. Marek oraz I. Rode-Babczenko w pracy [65], jednakże tej koncepcji nie będziemy tu rozważać.

4. SYSTEMY ROZPROSZONE

4.1. WPROWADZENIE

W praktyce spotykamy się często z następującą sytuacją: istnieje wiele różnych niezależnych systemów informacyjnych i w pewnym momencie chcemy je połączyć, np. w postaci sieci, w jeden duży system informacyjny. Na przykład w każdym mieście może istnieć oddział PZU posiadający w tym mieście system informacyjny o swych klientach. Centrala PZU może być zainteresowana połączeniem tych systemów w jeden system, tak aby mogła między systemami zachodzić wymiana informacji, a także aby władze centralne mogły otrzymać odpowiedzi na pytania dotyczące informacji w różnych systemach. Centrala PZU może być zainteresowana np. uzyskaniem informacji, w którym oddziale PZU jest największa liczba wypadków samochodowych o stratach powyżej 10000 zł, bądź też jaka jest najmniejsza średnia wypłacanych odszkodowań dla oddziału i które oddziały tę średnią przekraczają o określoną wartość itd. Może też istnieć nieco inna sytuacja. W danym mieście mogą istnieć systemy informacyjne różnych instytucji dotyczące mieszkańców tegoż miasta, takie jak system informacyjny dotyczący ubezpieczeń, lecznictwa, przestępstw kryminalnych itd. I tu również może zaistnieć potrzeba wymiany informacji między systemami, a także uzyskiwania informacji dotyczącej różnych aspektów opisanych w systemie obiektów. Na przykład możemy chcieć wiedzieć, które osoby będące w trakcie leczenia na określoną chorobę spowodowały wypadek samochodowy (może to być potrzebne dla stwierdzenia, czy określone leki nie wywołują osłabienia reakcji kierowcy). Informacje te znajdują się w dwu różnych systemach, medycznym i ubezpieczeniowym. Powstaje pytanie, czy jest tutaj możliwe sensowne utworzenie ze wszystkich tych systemów specjalistycznych jednego systemu informacyjnego, tak aby można w nim było w wygodny sposób uzyskiwać informacje zawarte w różnych systemach specjalistycznych.

Jeszcze inny przykład. W zakładzie produkcyjnym mogą istnieć systemy informacyjne, dotyczące personelu, materiałów, narzędzi itd. Również i tutaj celowe może okazać się traktowanie wszystkich tych systemów całościowo, tak aby w ten sposób rozszerzyć możliwości uzyskania informacji znajdującej się w różnych systemach.

We wszystkich wspomnianych przypadkach można systemy lokalne połączyć w postaci sieci, tworząc w ten sposób jeden system, który będziemy nazywać *rozproszonym systemem informacyjnym.* W systemach tego typu powstaje wiele problemów dotyczących np. metod wymiany informacji między poszczególnymi systemami, algorytmów jednoczesnej obsługi wielu użytkowników itd. Tutaj nie będziemy zajmować się tymi problemami, interesować nas będzie jeden ważny, zasadniczy problem: jaki jest mechanizm uzyskiwania odpowiedzi w systemach rozproszonych, a w szczególności struktura logiczna tego mechanizmu. Problem ten od tej strony nie był dostatecznie badany, a wydaje się, że jego dokładna analiza ma zasadnicze znaczenie dla budowy systemów rozproszonych.

Dodajmy, że w przypadku gdy nie istnieją jeszcze poszczególne lokalne systemy informacyjne i należy dopiero zaprojektować taki system od początku, powstaje problem, czy lepiej jest zbudować najpierw szereg systemów lokalnych, a następnie połączyć je w postaci sieci współpracujących systemów, czy też lepiej jest zbudować jeden system centralny zawierający wszystkie interesujące nas informacje, a wszystkim użytkownikom zapewnić tylko odpowiedni dostęp do informacji w systemie centralnym. Pełna analiza tego problemu jest dość złożona i tutaj zajmiemy się tylko pewnymi jego aspektami.

W systemach rozproszonych powstaje od razu problem ochrony informacji przed niepowołanymi osobami. Nie każda bowiem informacja może być dostępna dla każdego użytkownika systemu. Sprawą tą również nie będziemy się tu bliżej zajmować. Zainteresowanego Czytelnika odsyłamy do interesujących prac Michalewicza [73], [74], [75], [76], [77], dotyczących tego zagadnienia.

W pracy tej będziemy się zajmować systemami rozproszonymi, takimi, jak w pierwszych dwu przykładach. Zaproponowane bowiem podejście bardzo dobrze pozwala analizować systemy informacyjne tego właśnie rodzaju. Natomiast systemy takie, jak w trzecim przykładzie wygodniej badać za pomocą tzw. modelu relacyjnego. Szczegółowe dane na ten temat można znaleźć w pracy Hutta [18].

4.2. SYSTEM ROZPROSZONY I JEGO JĘZYKI

Definicja

Niech $S_1, S_2, \ldots, S_n$ będą systemami informacyjnymi, gdzie $S_i = \langle X_i, A_i, V_i, \varrho^i \rangle$. System S będący złożeniem systemów S_i, tj.

$$S = \bigcup_{i=1}^{n} S_i$$

i spełniającym warunki złączenia podane w p. 1.7, będziemy nazywać *systemem rozproszonym*, a systemy S_i — *składowymi systemu S*. Składowe systemu S są oczywiście jego podsystemami. ■

Niech L_S oznacza język systemu S, zaś L_{S_i} (lub krócej L_i) język podsystemu S_i.

Język L_S będziemy nazywać *językiem centralnym systemu S*, języki zaś L_{S_i} jego podsystemów składowych będziemy nazywali *językami lokalnymi systemu S*.

Główny problem, którym będziemy się zajmować w niniejszym rozdziale, jest następujący:

Dane jest pytanie w języku centralnym, tj. pytanie dotyczące informacji zawartej w kilku podsystemach, podać metodę, która pozwoli pytanie to zamienić na odpowiednie pytanie częściowe w językach lokalnych, takie że odpowiedzi na nie (odpowiedzi częściowe) pozwolą uzyskać całą odpowiedź, tj. odpowiedź na pytanie centralne. Mówiąc nieco inaczej, chodzi nam o podanie metody określenia znaczenia dowolnego termu $t \in L_S$ w systemie S w zależności od jego znaczeń w podsystemach S_i.

Sformułowanie to jest niezbyt ścisłe, gdyż term t może nie należeć do języka L_{S_i} — gdyż może on zawierać atrybuty występujące w języku L_S, lecz nie występujące w języku L_{S_i} — nie można więc w takim przypadku określić jego znaczenia w podsystemie S_i. Można jednak wtedy term t odpowiednio zmodyfikować, usuwając z niego „niepotrzebne" atrybuty i wtedy znaczenie tak zmodyfikowanego termu można ściśle zdefiniować. Problem ten wynika stąd, że zadający pytanie traktuje system jako całość i nie jest zainteresowany, czy system jest rozproszony, czy też nie. Fizycznie jednak jeden taki całościowy system nie istnieje, lecz składa się on z kilku połączonych ze sobą części składowych. Wobec tego nie jest możliwe uzyskanie bezpośrednie odpowiedzi, a odpowiedź można uzyskać przez złożenie odpowiedzi częściowych.

Na przykład, jeżeli składowymi podsystemami jakiegoś systemu rozproszonego są między innymi systemy: medyczny i ubezpieczeniowy, to odpowiedź na pytanie o zbiór osób, które są chore na określoną chorobę i spowodowały jednocześnie wypadek samochodowy, będzie iloczynem mnogościowym zbioru osób chorych na określoną chorobę i zbioru osób, które spowodowały wypadek samochodowy. Jedną część odpowiedzi na to pytanie uzyskuje się z systemu medycznego, drugą z ubezpieczeniowego, iloczyn tych odpowiedzi częściowych daje dopiero odpowiedź całkowitą.

Nie zawsze sytuacja jest taka prosta. Rozwiązanie tego problemu w ogólnym przypadku, tj. gdy system rozproszony składa się z dowolnych podsystemów składowych, nie jest wcale oczywiste i banalne, jak w rozpatrywanym przykładzie. Ponieważ problem ten jest skomplikowany i praktycznie ogólny przypadek nie jest zbyt interesujący, nie będziemy się nim tutaj zajmować — a rozpatrzymy dokładnie jedynie dwa szczególne przypadki tego zadania, które mają odniesienie do praktyki.

4.3. SYSTEM Z ROZPROSZONYMI ATRYBUTAMI

Pierwszy ze wspomnianych dwu przykładów systemów rozproszonych dotyczy sytuacji, w których wszystkie podsystemy systemu rozproszonego mają te same obiekty, natomiast różnią się atrybutami. Systemy takie będziemy nazywać *systemami z rozproszonymi atrybutami*. Sytuacja taka występuje np. wówczas, gdy w tym samym rejonie (województwie, mieście, zakładzie) znajdują się różne systemy informacyjne dotyczące tych samych obiektów (osób). Mogą to być np. systemy zawierające informacje kadrowe, medyczne, ubezpieczeniowe, kryminalne — dotyczące tej samej grupy ludności, np. w jednym dużym zakładzie pracy, okręgu, mieście itp. W przypadku gdy systemy te są połączone w jeden system, każdy z wymienionych podsystemów można traktować jako podsystem z ograniczonymi atrybutami systemu rozproszonego utworzonego z systemów składowych.

Określimy obecnie semantykę języka centralnego L_S. Przez $t|_{A_j}$, gdzie $t \in L_S$ oraz t jest termem prostym, oznaczymy term otrzymany z termu t przez usunięcie z t wszystkich deskryptorów zawierających atrybuty nie należące do A_j (A_j jest zbiorem atrybutów podsystemu S_j). Jeżeli t nie zawiera atrybutów należących do A_j, to $t|_{A_i} = 1$. Będziemy mówili, że term $t|_{A_j}$ jest *termem t zredukowanym do zbioru atrybutów* A_j.

Na przykład, jeżeli w systemie S występują atrybuty a, b, c, to następujący term prosty w języku L_S

$$(a, v)\cdot(b, u)\cdot(c, w),\; v \in V_a,\; u \in V_b,\; w \in V_c$$

zredukowany do zbioru atrybutów $\{a, c\}$ będzie miał postać

$$(a, v)\cdot(c, w)$$

Term prosty

(NAZWISKO = Kowalski) · (ADRES = Warszawa) · (OBYWATELSTWO = = polskie)

zredukowany do zbioru atrybutów {NAZWISKO, ADRES} będzie miał postać

(NAZWISKO = Kowalski) · (ADRES = Warszawa)

Niech $S = \bigcup_{i=1}^{n} S_i$ będzie systemem informacyjnym z rozproszonymi atrybutami oraz niech postacią normalną termu $t \in L_S$ będzie

$$t_1 + t_2 + \ldots + t_k$$

Łatwo sprawdzić, że dla każdego $t \in L_S$

$$\sigma_S(t) = \bigcup_{i=1}^{k} \bigcap_{j=1}^{n} \sigma_{S_j}(t_i|_{A_j})$$

W celu uzyskania odpowiedzi w języku centralnym w systemie z rozproszonymi atrybutami na pytanie t należy t sprowadzić najpierw do postaci normalnej, a następnie otrzymać odpowiedź na term t_i ograniczony do atrybutów A_j — w każdym podsystemie S_j. Iloczyn wszystkich odpowiedzi stanowi odpowiedź w systemie S na term prosty t_i. Suma wszystkich odpowiedzi na termy proste występujące w postaci normalnej termu t stanowi ostateczną odpowiedź na term t w systemie S.

Przykład 4.1

Rozpatrzmy prosty system z rozproszonymi atrybutami składający się z dwu podsystemów określonych tabelkami

System S_1

X	a	b
x_1	u_1	v_1
x_2	u_2	v_1
x_3	u_2	v_2
x_4	u_1	v_2

System S_2

X	b	c
x_1	v_1	w_2
x_2	v_1	w_2
x_3	v_2	w_1
x_4	v_2	w_1

Obliczymy znaczenie termu $t=(a, u_2)+(c, w_2)$ w systemie rozproszonym $S_1 \cup S_2$.

Postać normalna termu $t=(a, u_2)+(c, w_2)$ jest następująca:

$$t=(a, u_2)\cdot(b, v_1)\cdot(c, w_2)+(a, u_2)\cdot(b, v_2)\cdot(c, w_1)+$$
$$+(a, u_1)\cdot(b, v_1)\cdot(c, w_2)+(a, u_2)\cdot(b, v_1)\cdot(c, w_2)$$

Oznaczmy kolejno występujące termy proste w termie t przez t_1, t_2, t_3, t_4, oraz niech $A_1=\{a, b\}$ i $A_2=\{b, c\}$. Znaczenie odpowiednio ograniczonych termów prostych w systemach S_1, S_2 podano poniżej:

$$\sigma_{S_1}(t_1|_{A_1}) = \{x_2\}$$
$$\sigma_{S_1}(t_2|_{A_1}) = \{x_3\}$$
$$\sigma_{S_1}(t_3|_{A_1}) = \{x_1\}$$
$$\sigma_{S_1}(t_4|_{A_1}) = \{x_2\}$$
$$\sigma_{S_2}(t_1|_{A_2}) = \{x_1, x_2\}$$
$$\sigma_{S_2}(t_2|_{A_2}) = \{x_3, x_4\}$$
$$\sigma_{S_2}(t_3|_{A_2}) = \{x_1, x_2\}$$
$$\sigma_{S_2}(t_4|_{A_2}) = \{x_1, x_2\}$$

Odpowiednie iloczyny będą miały postać

$$\sigma_{S_1}(t_1|_{A_1}) \cap \sigma_{S_2}(t_1|_{A_2}) = \{x_2\} \cap \{x_1, x_2\} = \{x_2\}$$
$$\sigma_{S_1}(t_2|_{A_1}) \cap \sigma_{S_2}(t_2|_{A_2}) = \{x_3\} \cap \{x_3, x_4\} = \{x_3\}$$
$$\sigma_{S_1}(t_3|_{A_1}) \cap \sigma_{S_2}(t_3|_{A_2}) = \{x_1\} \cap \{x_1, x_2\} = \{x_1\}$$
$$\sigma_{S_1}(t_4|_{A_1}) \cap \sigma_{S_2}(t_4|_{A_2}) = \{x_2\} \cap \{x_1, x_2\} = \{x_2\}$$

Sumując otrzymane iloczyny jako ostateczny rezultat, otrzymamy

$$\sigma_S(t) = \{x_1, x_2, x_3\}$$

W rozpatrywanym przykładzie wartość termu $t=(a, u_2)+(c, w_2)$ moglibyśmy również obliczyć bez sprowadzania go do postaci normalnej, obliczając znaczenie składnika (a, u_2) w systemie S_1 oraz składnika (c, w_2) — w systemie S_2, tj. w systemach, w których występują atrybuty, odpowiednio, a i c.

Dla pytań postaci $t = t_1 + t_2 + \ldots + t_k$, gdzie t_i są iloczynami deskryptorów (niekoniecznie termami prostymi), wartość t w systemie S można obliczyć bez sprowadzania t do postaci normalnej w S — obliczając od razu wartości odpowiednio zredukowanych termów t_i w systemach składowych, a następnie obliczając ich iloczyny i sumę. Szczegóły takiego skróconego obliczania można znaleźć w pracy [5]. Tak więc, gdybyśmy chcieli obliczyć wartości termu $(a, u_2) \cdot (c, w_2)$ zamiast sprowadzać go najpierw do postaci normalnej, a następnie obliczyć jego wartość według podanego wzoru, możemy bezpośrednio obliczyć najpierw wartość deskryptora (a, u_2) w systemie S_1, otrzymując jako wynik $\{x_2, x_3\}$, a następnie wartość deskryptora (c, w_2) w systemie S_2, otrzymując $\{x_1, x_2\}$. W rezultacie wartością termu $(a, u_2) \cdot (c, w_2)$ w systemie $S_1 \cup S_2$ jest zbiór $\{x_2\}$. ■

Przykład 4.2

Rozpatrzmy bardziej realistyczny przykład instytucji handlu zagranicznego, która ma dwa systemy informacyjne, jeden dotyczący danych personalnych, drugi — wyjazdów zagranicznych pracowników tej instytucji. Przyjmiemy, że w systemie informacji personalnej występują następujące atrybuty:

NAZWISKO

WIEK

WYKSZTAŁCENIE

ADRES

ODDZIAŁ

STANOWISKO

STAŻ

W systemie informacyjnym o wyjazdach natomiast występują atrybuty następujące:

NAZWISKO

NR PASZPORTU

NR WYJAZDU

KRAJ

CEL WYJAZDU

DATA WYJAZDU

DATA POWROTU

Gdybyśmy chcieli wyszukać wszystkie osoby z wyższym wykształceniem w wieku poniżej 30 lat, które wyjeżdżały w lipcu 1981 r. do Bułgarii w sprawie zawarcia kontraktu na zakup pomidorów, należałoby najpierw w systemie informacji personalnej wyszukać osoby określone termem

(WYKSZTAŁCENIE = Wyższe) · (WIEK < 30)

a następnie w systemie informacyjnym o wyjazdach wyszukać osoby według termu

(KRAJ = Bułgaria) · (DATA WYJAZDU = lipiec 1981) · (CEL WYJAZDU = = zakup pomidorów)

Iloczyn dla tak otrzymanych zbiorów da nam żądaną odpowiedź. ■

4.4. SYSTEM Z ROZPROSZONYMI OBIEKTAMI

Drugi dyskutowany tu przypadek systemów rozproszonych odpowiada sytuacji, w której wszystkie systemy składowe systemu rozproszonego mają jednakowe atrybuty, ale różne zbiory obiektów. Przykładem takiego systemu może być system informacji personalnej dużego przedsiębiorstwa, mającego oddziały w różnych miejscowościach. Atrybuty w takim przypadku będą oczywiście w każdym podsystemie jednakowe, natomiast zbiór obiektów — będzie różny. Dopuszczamy tu możliwość, że niektóre osoby mogą pracować jednocześnie w kilku oddziałach (np. na niepełnym etacie). Systemy tego rodzaju będziemy nazywać *systemami z rozproszonymi obiektami.*

Semantyka języka centralnego takiego systemu jest określona następująco:

Niech $S = \bigcup_{i=1}^{n} S_i$ będzie systemem informacyjnym z rozproszonymi obiektami i niech t będzie termem należącym do L_S, którego postacią normalną w L_S jest $t_1 + t_2 + \ldots + t_k$.

Znaczenie termu t w L_S jest określone wzorem

$$\sigma_S(t) = \bigcup_{i=1}^{k} \bigcup_{j=1}^{n} \sigma_{S_j}(t_i) = \bigcup_{j=1}^{n} \sigma_{S_j}(t)$$

Obliczanie odpowiedzi w języku centralnym z rozproszonymi obiektami na pytanie t polega więc najpierw na sprowadzeniu pytania t do postaci normalnej, a następnie na obliczeniu odpowiedzi na każdy term elementarny i zsumowaniu wszystkich tak otrzymanych odpowiedzi częściowych, lub krócej — należy obliczyć wartość termu t w każdym podsystemie i zsumować tak otrzymane wyniki. Ponieważ wszystkie podsystemy mają jednakowe atrybuty, postępowanie takie jest dopuszczalne.

Przykład 4.3

Rozpatrzmy system rozproszony składający się z dwu podsystemów S_1 i S_2 określonych tabelkami:

System S_1

X	a	b	c
x_1	u_1	v_1	w_2
x_2	u_2	v_1	w_1
x_3	u_2	v_2	w_1
x_4	u_1	v_2	w_1

System S_2

Y	a	b	c
y_1	u_2	v_1	w_1
y_2	u_2	v_2	w_1
y_3	u_1	v_2	w_2
y_4	u_1	v_1	w_1

Dla uzyskania odpowiedzi na pytanie $(a, u_1)+(c, w_2)$ należy uzyskać najpierw odpowiedź na nie w systemie S_1, a następnie w systemie S_2. W pierwszym przypadku otrzymamy $\{x_1, x_4\}$, w drugim zaś — $\{y_3, y_4\}$. Całkowita odpowiedź będzie więc miała postać $\{x_1, x_4, y_3, y_4\}$. ■

4.5. UŻYTKOWNIK CENTRALNY A UŻYTKOWNIK LOKALNY

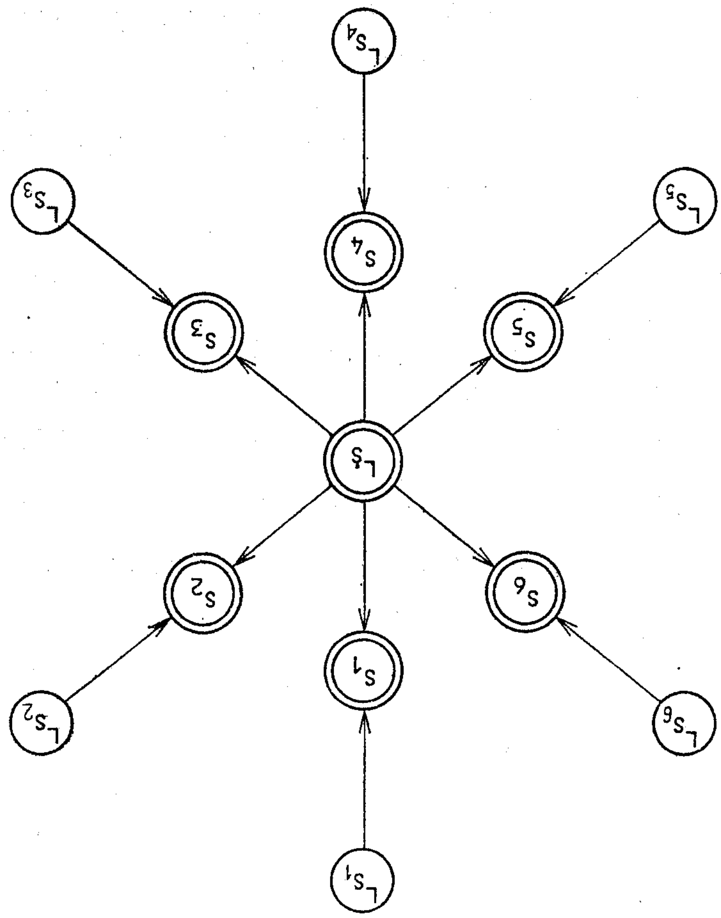
W dotychczasowych rozważaniach nie sprecyzowaliśmy dokładnie, kto w systemie rozproszonym jest użytkownikiem języka centralnego, a kto lokalnego.

W przypadku, gdy składnikami systemu rozproszonego są na przykład systemy: personalny, kryminalny, medyczny — jest rzeczą oczywistą, że użytkownik systemu medycznego nie powinien mieć dostępu do danych w systemie kryminalnym, natomiast użytkownik systemu kryminalnego może mieć dostęp do wszystkich podsystemów. Pomijamy tu sprawę tajemnicy lekarskiej.

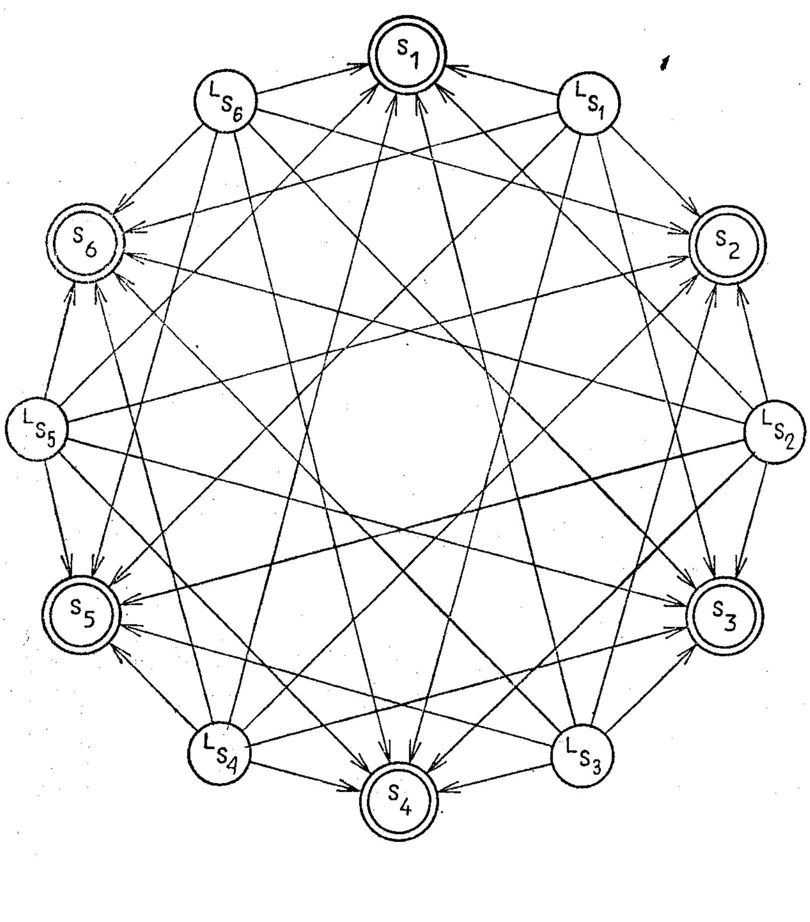
W systemie informacyjnym koncernu składającego się z wielu przedsiębiorstw zarząd koncernu ma oczywiście dostęp do wszystkich podsystemów przedsiębiorstwa, natomiast przedsiębiorstwa mogą mieć w zasadzie dostęp tylko do danych w swoim systemie.

W przypadku systemu rozproszonego, którego składnikami są np. systemy biblioteczne, każdy użytkownik podsystemu może mieć dostęp do wszystkich pozostałych podsystemów. Szukając bowiem określonej książki, czy też części zamiennej, chcielibyśmy mieć możliwość sprowadzenia jej z dowolnej biblioteki lub magazynu, jeśli w tym, w którym jej aktualnie szukamy, jej nie ma. W konsekwencji z tego właśnie powodu rozproszone systemy informacyjne można podzielić na dwie klasy: do pierwszej należą systemy, w których istnieje wyróżniony użytkownik (centralny lub użytkownicy) mający dostęp do wszystkich podsystemów systemu rozproszonego i dysponujący centralnym językiem, przy czym użytkownicy podsystemów (lokalni) mają dostęp tylko do własnych podsystemów; do drugiej klasy należą systemy, w których nie ma wyróżnionego użytkownika centralnego, a każdy z użytkowników lokalnych ma dostęp do dowolnych podsystemów systemu rozproszonego.

Teraz można już wprowadzić pojęcia systemu informacyjnego z rozproszonym i centralnym dostępem. Przez *system informacyjny z centralnym dostępem* rozumiemy system, w którym dostęp do wszystkich podsystemów ma specjalnie wyróżniony



Rys. 4.1. System z centralnym dostępem



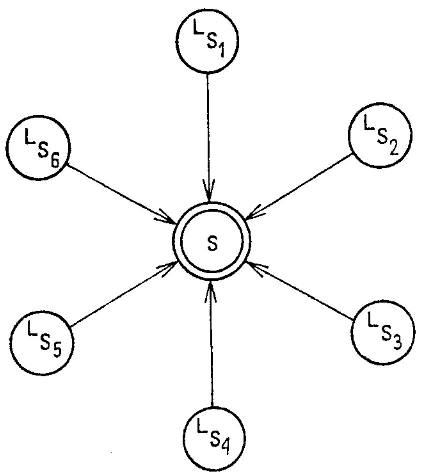
Rys. 4.2. System z rozproszonym dostępem

użytkownik bądź grupa użytkowników; natomiast w *systemie z rozproszonym dostępem* dostęp do każdego podsystemu ma każdy użytkownik systemu.

Oba rodzaje systemów przedstawiono umownie na rysunkach 4.1 i 4.2. Połączenia między podsystemami wskazują dostęp do informacji dla użytkowników systemu.

4.6. PAMIĘĆ CENTRALNA I PAMIĘĆ LOKALNA

Rozpatrywane do tej pory systemy rozproszone składały się z lokalnych systemów informacyjnych, połączonych odpowiednio ze sobą. Istnieje również inna możliwość zrealizowania systemów rozproszonych. Można mianowicie zamiast wielku lokalnych systemów $S_1, S_2, \ldots, S_n$ tworzących system rozproszony $S = \bigcup_{i=1}^{n} S_i$ zrealizować fizycznie jeden system S ze wspólną pamięcią danych, jak to pokazano na rys. 4.3.



Rys. 4.3. System rozproszony z centralną pamięcią

W przypadku gdy składniki systemu S są podsystemami z ograniczonymi atrybutami systemu S, każdy użytkownik może używać języka, w którym występują atrybuty dozwolone dla niego, natomiast gdy składniki S_i systemu S są podsystemami z ograniczonymi obiektami systemu S, każdy użytkownik ma jednakowy język centralny — natomiast ma ograniczony dostęp do obiektów systemu S.

Mówiąc inaczej, w obu przypadkach systemów rozproszonych zamiast wielu systemów lokalnych tworzących razem system rozproszony można zbudować jeden system centralny, w którym zależnie od jego rodzaju użytkownicy lokalni mają ograniczone możliwości w używaniu atrybutów w swych zapytaniach, bądź też mogą oni używać wszystkich atrybutów, lecz mają ograniczony dostęp do obiektów systemu.

Mówiąc jeszcze inaczej, w pierwszym przypadku systemu rozproszonego użytkownicy lokalni mogą otrzymywać tylko niektóre informacje o każdym obiekcie systemu — w drugim zaś przypadku użytkownicy mogą otrzymać każdą informację, ale nie o wszystkich obiektach. Możemy więc mówić tu o systemach z pamięcią centralną i rozproszoną (lokalną).

Algorytmy szukania odpowiedzi w systemach z pamięcią centralną różnią się nieco od algorytmów w systemach z pamięcią rozproszoną. Jeżeli $S = \bigcup_{i=1}^{n} S_i$ jest systemem z rozproszonymi atrybutami oraz centralną pamięcią, to odpowiedź na każde pytanie $t \in L_{S_i}$ będzie oczywiście sumą odpowiednich zbiorów elementarnych relacji $\underset{S}{\sim}$. Jednakże algorytm oparty bezpośrednio na tym fakcie byłby wysoce nieefektywny. Znacznie wygodniejsze i efektywniejsze byłoby obliczanie odpowiedzi na pytanie $t \in L_{S_i}$ na podstawie relacji $\underset{S_i}{\sim}$, tj. przez sumowanie odpowiednich zbiorów elementarnych relacji $\underset{S_i}{\sim}$. Tak więc w tym drugim przypadku powinniśmy zbiór obiektów X systemu S podzielić na zbiory elementarne zgodnie z relacjami $\underset{S_1}{\sim}, \underset{S_2}{\sim}, \ldots, \underset{S_n}{\sim}$.

Zadane pytanie $t \in L_{S_i}$ należy sprowadzić do postaci normalnej w języku L_{S_i}, a następnie wyszukać odpowiednie zbiory elementarne relacji $_{S_i}$, ściślej, jeżeli $t \in L_{S_i}$, to

$$\sigma_{S_i}(t) = \bigcup_{j=1}^{k} \sigma_{S_i}(t_j)$$

gdzie postacią normalną termu t w języku L_{S_i} jest $t_1 + t_2 + \ldots + t_k$.

Jeżeli natomiast $t \in L_S$, to

$$\sigma_S(t) = \bigcup_{j=1}^{l} \sigma_S(t_j)$$

gdzie postacią normalną termu t w języku L_S jest $t_1 + t_2 + \ldots, t_l$.

Odpowiedzi na pytania użytkownika lokalnego podsystemu S_i są sumą zbiorów elementarnych relacji $\underset{S_i}{\sim}$, natomiast odpowiedzi użytkownika centralnego są sumami zbiorów elementarnych relacji $\underset{S}{\sim}$.

Algorytm otrzymywania odpowiedzi w systemie z centralną pamięcią jest niewątpliwie prostszy niż algorytm w systemie z rozproszoną pamięcią, nie można tu bowiem wykonywać operacji mnożenia mnogościowego zbiorów, która nie jest zbyt wygodna do realizacji — a całą odpowiedź otrzymuje się wyłącznie przez sumowanie zbiorów.

Jeżeli $S = \bigcup_{i=1}^{n} S_i$ jest systemem z rozproszonymi obiektami oraz centralną pamięcią, to algorytm szukania odpowiedzi jest tutaj również różny od tego, który podaliśmy dla pamięci rozproszonej. Odpowiedź na pytanie t zadane przez użytkownika podsystemu S_i będzie sumą zbiorów elementarnych relacji $\underset{S_i}{\sim}|_{X_i}$, a więc relacji $\underset{S_i}{\sim}$ obciętej do zbioru X_i. Oznacza to, że w tego rodzaju systemie rozproszonym obiekty X systemu S powinny być podzielone na zbiory elementarne relacji $\underset{S}{\sim}$, a następnie zbiory elementarne relacji $\underset{S}{\sim}$ powinny być podzielone na zbiory elementarne relacji $\underset{S_1}{\sim}, \underset{S_2}{\sim}, \ldots, \underset{S_n}{\sim}$.

Algorytm otrzymania odpowiedzi na pytanie t postawione przez użytkownika systemu S_i sprowadza się więc tutaj do przekształcenia pytania t do postaci normalnej, a następnie wybrania tych fragmentów odpowiednich zbiorów elementarnych relacji $\underset{S}{\sim}$, które stanowią zbiory elementarne relacji $\underset{S_i}{\sim}$.

Pisząc ściślej

$$\sigma_{S_i}(t) = \bigcup_{j=1}^{k} \sigma_{S_i}(t_j)$$

gdzie term t ma postać normalną $t_1 + t_2 + \ldots + t_k$ oraz t jest zadane przez użytkownika systemu S_i.

Natomiast jeżeli pytanie t jest zadane przez użytkownika centralnego, to

$$\sigma_S(t) = \bigcup_{j=1}^{k} \sigma_S(t_j)$$

Mówiąc inaczej, odpowiedzi na pytanie użytkownika centralnego będą sumami zbiorów elementarnych relacji $\underset{S}{\sim}$, natomiast odpowiedzi na pytania stawiane przez użytkownika lokalnego będą sumami odpowiednich fragmentów zbiorów elementarnych relacji $\underset{S}{\sim}$.

W przypadku systemów rozproszonych z pamięcią centralną zawsze odpowiedzi uzyskujemy jako sumę odpowiednich zbiorów elementarnych. Wybór pamięci centralnej bądź lokalnej zależy jeszcze od wielu innych czynników, których tutaj nie będziemy dyskutować.

Przytoczone rozważania miały na celu jedynie zbadanie podstawowych własności systemów rozproszonych z punktu widzenia struktury algorytmu wyszukiwania odpowiedzi.

4.7. UWAGI KOŃCOWE

Jak wynika z rozważań przeprowadzonych w tym rozdziale, pojęcie rozproszonego systemu informacyjnego nie jest jednoznaczne. Można wyróżnić co najmniej trzy aspekty, z punktu widzenia których mówimy o rozproszeniu systemu:

(1) język,

(2) obiekty,

(3) pamięć.

W pierwszym przypadku mamy do czynienia z faktem „podziału" języka między użytkowników, tj. udostępnienia każdemu użytkownikowi fragmentu języka (w szczególności całego języka) — zgodnie z jego potrzebami.

W drugim przypadku są „podzielone" między użytkowników obiekty systemu, tzn. każdy użytkownik ma dostęp do informacji nie o wszystkich obiektach, lecz tylko o tych, które są mu „przypisane" (w szczególności mogą to być wszystkie obiekty).

W ostatnim przypadku mamy do czynienia z „podziałem" pamięci między użytkowników. Każdy z nich może mieć swój podsystem bądź też może istnieć tylko jeden centralny system, z którego korzystają wszyscy użytkownicy z odpowiednimi ograniczeniami. Oczywiście mogą też istnieć systemy „mieszane", w których jednocześnie występują różne aspekty „rozproszenia".

W rozdziale tym podaliśmy jedynie szkic jednego podstawowego problemu w systemach rozproszonych, a mianowicie przeanalizowaliśmy niektóre mechanizmy algorytmów wyszukiwania odpowiedzi, nie wyczerpując jednak tego zagadnienia.

Istnieje bogata literatura dotycząca systemów rozproszonych, jednak jak się wydaje, nie wyczerpuje ona w pełni tematu. Jeśli chodzi o analizę dyskutowanych tu mechanizmów odpowiedzi, jest ona bardzo uboga i chyba dalece niewystarczająca, zarówno z punktu widzenia podstaw teoretycznych, jak i praktycznych zastosowań. Dlatego literatury tej nie cytujemy.

Warto dodać, że problem ten jest bardzo ważny praktycznie, chociażby z uwagi na badanie sieci komputerowej, gdzie poruszane tu problemy mają charakter zasadniczy.

W pracach dotyczących sieci komputerowych bada się głównie aspekt komunikacyjny między poszczególnymi komputerami, nie są jednak analizowane do tej pory wystarczająco mechanizmy szukania odpowiedzi w takich sieciach — a wydaje się, że jest to sprawa podstawowa. Gdy nie znamy bowiem dostatecznie tych mechanizmów, może się zdarzyć, że wyszukiwane odpowiedzi są bądź nieprawidłowe, bądź jeśli nawet są one prawidłowe można je otrzymać znacznie prostszą drogą.

5. SYSTEMY INFORMACYJNE WIELOWARTOŚCIOWE

5.1. WPROWADZENIE

W omawianych do tej pory systemach informacyjnych przyjmowaliśmy, że każdy atrybut może przyjmować dla dowolnego obiektu dokładnie jedną spośród możliwych wartości. W rzeczywistości nie zawsze warunek ten jest spełniony. Mogą istnieć takie atrybuty, które nie spełniają tego warunku. Na przykład w systemie informacyjnym o wyrobach atrybut MATERIAŁ może oznaczać wszystkie rodzaje materiałów użytych do wykonania danego wyrobu. Wtedy wartością atrybutu MATERIAŁ może być np. zbiór {żelazo, mosiądz, miedź}. A więc wartością tego atrybutu może być dowolny podzbiór zbioru możliwych wartości, a nie tylko jeden element tego zbioru. Podobnie wartością atrybutu KOOPERANT może być wiele instytucji współpracujących w produkcji danego wyrobu. Takie same własności mają atrybuty DOSTAWCA i ODBIORCA. Wartościami tych atrybutów są odpowiednio wszystkie firmy dostarczające lub zakupujące dany wyrób. Atrybutem takim jest również atrybut AUTOR w systemie bibliotecznym, gdyż jakaś publikacja może mieć więcej niż jednego autora. Innym jeszcze przykładem atrybutu o takich własnościach może być w personalnym systemie informacyjnym atrybut JĘZYK. W rubryce tej wymieniamy wszystkie języki, które zna dana osoba. Atrybuty takie, które mogą przybierać jednocześnie kilka wartości, będziemy nazywali *atrybutami wielowartościowymi* — a systemy informacyjne zawierające atrybuty wielowartościowe — nazwiemy *systemami informacyjnymi wielowartościowymi*. Tak więc w systemie informacyjnym wielowartościowym mogą występować zarówno atrybuty jednowartościowe, jak i wielowartościowe. Powstaje pytanie, czy wprowadzone w poprzednich rozdziałach pojęcie systemu informacyjnego oraz języka można uogólnić na przypadek systemów wielowartościowych.

W rozdziale tym zajmiemy się dokładną analizą tego problemu. Zanim jednak do tego przystąpimy, rozpatrzmy nieco dokładniej przykład systemu wielowartościowego.

5.2. PRZYKŁAD SYSTEMU WIELOWARTOŚCIOWEGO

Rozpatrzmy przykład systemu informacyjnego, w którym występuje atrybut wielowartościowy JĘZYK — oznaczający znajomość wszystkich języków obcych przez każdą osobę figurującą w systemie. Jakie rodzaje pytań można wtedy stawiać odnośnie do znajomości języków? Można pytać o wszystkie osoby znające określony język, np. angielski. Może się zdarzyć, że osoby te znają również jednocześnie inne języki. Można więc pytać o osoby znające tylko język angielski i nie znające żadnych innych języków. Można również pytać o osoby, które znają jednocześnie dwa określone języki, np. angielski i niemiecki — a wśród nich o osoby znające dokładnie tylko te dwa języki i żadnych innych. Możliwości jest tu bardzo dużo i są one bardziej złożone niż w systemach jednowartościowych.

Na przykład odpowiedź na pytanie

(JĘZYK = polski)

może być różna: może być nią zbiór wszystkich osób znających język polski, a może też być zbiór osób znających tylko język polski. Odpowiedź na pytanie

(JĘZYK = angielski, niemiecki)

może mieć jeszcze więcej wariantów. Przez odpowiedź na to pytanie możemy rozumieć np. zbiór tych osób, które znają jednocześnie te dwa języki i tylko te języki. Możemy również przyjąć, że odpowiedź na to pytanie stanowi zbiór osób, które znają co najmniej jeden z tych języków i jednocześnie nie znają żadnego innego języka. Pytanie takie można również interpretować w ten sposób, że odpowiedzią na nie będzie zbiór tych osób, które jednocześnie znają co najmniej te dwa języki. Wreszcie pytanie to można rozumieć jako pytanie o osoby znające co najmniej jeden z wymienionych języków.

Dla dokładnego zbadania tych sytuacji musimy najpierw sprecyzować pojęcie systemu informacyjnego wielowartościowego oraz związanego z nimi języka zapytań, tj. jego składni i semantyki — podobnie jak to uczyniliśmy dla systemów jednowartościowych.

5.3. DEFINICJA SYSTEMU INFORMACYJNEGO WIELOWARTOŚCIOWEGO

Definicja

Wielowartościowym systemem informacyjnym będziemy nazywać czwórkę

$$S = \langle X, A, V, \varrho \rangle$$

gdzie: X — zbiór obiektów, A — zbiór atrybutów,

$V = \bigcup_{a \in A} V_a$, V_a — dziedzina atrybutu a,

$\varrho\colon X \times A \to \mathscr{P}(V) - \{\emptyset\}$ — funkcja całkowita

i $\varrho(x, a) \subset V_a$ dla każdego $x \in X$, $a \in A$ ∎

Definicja systemu wielowartościowego różni się od definicji systemu jednowartościowego jedynie tym, że funkcją ϱ przypisuje obiektom i atrybutom nie pojedyncze wartości, lecz ich zbiory. Przyjmujemy, że funkcja ϱ podobnie jak poprzednio, jest całkowita, co oznacza, że żądamy, aby była ona zawsze określona. Nie dopuszczamy więc sytuacji, w której wartość atrybutu jest nieokreślona.

Definicja

Informacją o obiekcie x w systemie S jest funkcja ϱ_x określona w następujący sposób:

$$\varrho_x: A \to \mathscr{P}(V)$$

przy czym

$$\varrho_x(a) = \varrho(x, a) \subset V_a$$ ■

Definicja

Obiekty $x, y \in X$ są *równoważne w systemie S ze względu na atrybut a* (symbolicznie $x \underset{a}{\sim} y$) wtedy i tylko wtedy, gdy

$$\varrho_x(a) = \varrho_y(a)$$ ■

Definicja

Obiekty $x, y \in X$ są *równoważne w systemie S ze względu na podzbiór $B \subset A$* (symbolicznie $x \underset{B}{\sim} y$) wtedy i tylko wtedy, gdy

$$\varrho_x(a) = \varrho_y(a) \quad \text{dla każdego } a \in B.$$

W przypadku gdy $B = A$ będziemy mówili, że obiekty $x, y \in X$ są *równoważne w systemie S* i zapiszemy $x \underset{S}{\sim} y$, lub krótko $x \sim y$, gdy system S jest ustalony. ■

Oczywiście $x \underset{S}{\sim} y$ wtedy i tylko wtedy, gdy $\varrho_x = \varrho_y$.

Relacje $\underset{a}{\sim}$, $\underset{S}{\sim}$ są relacjami równoważności, gdyż

(1) $\varrho_x(a) = \varrho_x(a)$

(2) jeżeli $\varrho_x(a) = \varrho_y(a)$, to $\varrho_y(a) = \varrho_x(a)$,

(3) jeżeli $\varrho_x(a) = \varrho_y(a)$ oraz $\varrho_y(a) = \varrho_z(a)$, $\varrho_x(a) = \varrho_z(a)$

oraz

(4) $\varrho_x = \varrho_x$

(5) jeżeli $\varrho_x = \varrho_y$, to $\varrho_y = \varrho_x$

(6) jeżeli $\varrho_x = \varrho_y$ oraz $\varrho_y = \varrho_z$, to $\varrho_x = \varrho_z$.

Własności (1)÷(6) są analogiczne jak w przypadku systemów jednowartościowych. Każdy podzbiór atrybutów systemu generuje odpowiadającą mu relację równoważności. Klasy abstrakcji tej relacji równoważności zawierają wszystkie obiekty systemu nierozróżnialne w nim ze względu na te atrybuty. Zachodzi również następujący fakt:

$$\underset{S}{\sim} = \bigcap_{a \in A} \underset{a}{\sim}$$

Moglibyśmy więc traktować systemy wielowartościowe jak systemy jednowartościowe, w których każdy podzbiór wartości atrybutu a traktuje się jak pojedynczą wartość. To znaczy, jeżeli V_a jest zbiorem wartości atrybutu a, to atrybut a przybiera w systemie wielowartościowym $2^{\mathrm{card}(V_a)}$ wartości, tyle bowiem jest różnych kombinacji wartości atrybutu a. W konsekwencji wszystkie własności systemów jednowartościowych można interpretować w systemach wielowartościowych. Nie będziemy tu jednak rozwijać tej myśli, polecając jej kontynuację zainteresowanemu Czytelnikowi jako ćwiczenie (np. przebadać zależność atrybutów i pojęcie reduktu w systemach wielowartościowych) — wykorzystamy jedynie te elementy, które będą nam potrzebne do dalszych rozważań.

Proste przykłady zilustrują wprowadzone pojęcie systemu wielowartościowego.

Przykład 5.1

Niech S będzie systemem informacyjnym wielowartościowym z dwoma atrybutami a, b, gdzie a jest atrybutem wielowartościowym, b zaś jest atrybutem jednowartościowym, o następujących dziedzinach $V_a = \{u, v, w\}$, $V_b = \{p, q\}$. System ten określimy tabelką

X	a	b
x_1	u	p
x_2	u, v	q
x_3	u	q
x_4	v, w	q
x_5	v	p
x_6	v	p
x_7	u, v	q
x_8	v, w	p
x_9	u, v, w	q
x_{10}	w	p

Zbiory elementarne generowane przez atrybut a są następujące:

$$E_u = \{x_1, x_3\}, \quad E_{u,v} = \{x_2, x_7\}, \quad E_{u,v,w} = \{x_9\}$$
$$E_v = \{x_5, x_6\}, \quad E_{v,w} = \{x_4, x_8\},$$
$$E_w = \{x_{10}\}, \quad E_{u,w} = \emptyset,$$

Zbiory elementarne generowane przez atrybut b mają postać

$$E_p = \{x_1, x_5, x_6, x_8, x_{10}\}$$
$$E_q = \{x_2, x_3, x_4, x_7, x_9\}$$

Wyrażenie E_U jest skrótem wyrażenia $E_{(a,U)}$, które oznacza zbiór elementarny

$$E_{(a,U)}\{ = x \in X: \varrho_x(a) = U\}$$

Ogólnie

$$X_t = \{x: x \in \sigma_S(t)\}$$

Zbiorami elementarnymi całego systemu S są

$$E_u \cap E_p = \{x_1\} \qquad E_u \cap E_q = \{x_3\}$$
$$E_v \cap E_p = \{x_5, x_6\} \qquad E_{u,\,v} \cap E_q = \{x_2, x_7\}$$
$$E_w \cap E_p = \{x_{10}\} \qquad E_{v,\,w} \cap E_q = \{x_4\}$$
$$E_{v,\,w} \cap E_p = \{x_8\} \qquad E_{u,\,v,\,w} \cap E_q = \{x_9\}$$

■

Przykład 5.2

Podamy teraz mniej formalny przykład, który pozwoli dokładniej zrozumieć ideę wielowartościowości atrybutu. Rozpatrzmy wielowartościowy system informacyjny z jednym tylko atrybutem JĘZYK, w którym znajduje się informacja o znajomości języków obcych każdej osoby figurującej w tym systemie. Będziemy rozpatrywać tylko cztery języki: angielski, francuski, polski, węgierski, skrótowo oznaczone an, fr, pl, wg. Dziedziną atrybutu JĘZYK jest więc zbiór {an, fr, pl, wg}. Dla uproszczenia nie zdefiniujemy całego systemu informacyjnego, lecz podamy jedynie jego zbiory elementarne, które jak poprzednio, oznaczymy literą E ze wskaźnikiem u dołu wskazującym, jakie języki znają osoby należące do tego zbioru elementarnego. Na przykład E_{pl} oznacza zbiór osób mówiących tylko po polsku, zbiór $E_{\mathrm{an,\,pl}}$ oznacza zbiór osób znających język angielski i polski i tylko te dwa języki itd.

Przyjmijmy, że w rozpatrywanym systemie jest 10 niepustych zbiorów elementarnych:

$$E_{\mathrm{pl}}, E_{\mathrm{an}}, E_{\mathrm{an,fr}}, E_{\mathrm{an,wg}}, E_{\mathrm{fr,wg}}, E_{\mathrm{pl,wg}}, E_{\mathrm{an,pl}}, E_{\mathrm{an,wg,fr}}, E_{\mathrm{an,fr,pl}}, E_{\mathrm{an,fr,pl,wg}}.$$

Podamy teraz przykłady zbiorów osób w zależności od ich znajomości języków.

Zbiór wszystkich osób mówiących po polsku:

$$E_{\mathrm{pl}} \cup E_{\mathrm{pl,wg}} \cup E_{\mathrm{an,pl}} \cup E_{\mathrm{an,fr,pl}} \cup E_{\mathrm{an,fr,pl,wg}}$$

Zbiór wszystkich osób mówiących po angielsku:

$$E_{\mathrm{an}} \cup E_{\mathrm{an,fr}} \cup E_{\mathrm{an,wg}} \cup E_{\mathrm{an,pl}} \cup E_{\mathrm{an,wg,fr}} \cup E_{\mathrm{an,fr,pl}} \cup E_{\mathrm{an,fr,pl,wg}}$$

Zbiór wszystkich osób mówiących tylko jednym z języków, polskim lub angielskim i nie znających żadnych innych języków:

$$E_{\mathrm{an}} \cup E_{\mathrm{pl}} \cup E_{\mathrm{an,pl}}$$

Zbiór osób znających jednocześnie język polski i angielski:

$$E_{\mathrm{an,pl}} \cup E_{\mathrm{an,fr,pl}} \cup E_{\mathrm{an,fr,pl,wg}}$$

Zbiór osób nie znających języka angielskiego:

$$E_{\mathrm{pl}} \cup E_{\mathrm{fr,wg}} \cup E_{\mathrm{pl,wg}}$$

Zbiór osób znających jednocześnie język polski i angielski, ale nie znających języka węgierskiego:

$$E_{\mathrm{an,pl}} \cup E_{\mathrm{an,fr,pl}}$$ ■

Przykład ten daje dobre pojęcie o pytaniach, jakie można stawiać w systemach wielowartościowych.

5.4. SKŁADNIA JĘZYKA SYSTEMÓW WIELOWARTOŚCIOWYCH

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym wielowartościowym oraz L_S niech oznacza jego język.

Termy języka L_S będziemy budować z następujących symboli:

(a) stałych 0, 1,

(b) deskryptorów (a, U), $U \subset V_a$, $a \in A$,

(c) symboli operacji $\sim$, $+$, $\cdot$, $\rightarrow$, $\leftrightarrow$,

(d) symboli operacji $\underline{C}$, C, $\overline{C}$.

Termy języka L_S zdefiniujemy indukcyjnie w następujący sposób:

(1) stałe 0, 1 są termami;

(2) termem jest każdy deskryptor (a, U), dla $a \in A$, $U \subset V_a$

(3) jeżeli t, t' są termami, to również termami są następujące wyrażenia $(t + t')$, $(t \cdot t')$, $(t \rightarrow t')$, $(t \leftrightarrow t')$, $\sim t$;

(4) jeżeli t jest termem, to termami są również $\underline{C}t$, Ct, $\overline{C}t$.

Proste przykłady termów w języku systemów wielowartościowych podano poniżej:

$C(a, v) + C(a, U)$

$\sim [\underline{C}(a, v) \cdot \overline{C}(b, U)] + (c, W)$

$(a, V) + C(b, U) \cdot \underline{C}(b, W)$

Język ten różni się od języka systemów jednowartościowych jedynie tym, że występują w nim operatory $\underline{C}$, C, $\overline{C}$. Oczywiście deskryptory języka wielowartościowego mogą zawierać więcej niż jedną wartość atrybutu, jednakże w istocie atrybuty wielowartościowe można traktować jako takie atrybuty jednowartościowe, w których każdy podzbiór wartości atrybutu a traktujemy jako jedną wartość.

5.5. SEMANTYKA JĘZYKA SYSTEMÓW WIELOWARTOŚCIOWYCH

Wyrażenia języka systemów wielowartościowych oznaczają, podobnie jak w poprzednich przypadkach, podzbiory obiektów systemu, a więc dla określenia ich znaczenia wprowadzimy, tak jak poprzednio, funkcję

$$\sigma_S\colon L_S \rightarrow \mathscr{P}(X)$$

określającą znaczenie wyrażeń naszego języka, którą będziemy nazywali *semantyką języka* L_S.

Niech $S = \langle X, A, V, \varrho \rangle$ będzie wielowartościowym systemem informacyjnym. Funkcja σ dla języka tego systemu musi mieć następujące własności:

(1) $\sigma(0) = \emptyset, \qquad \sigma(1) = X$

(2) $\sigma(a, U) = \{x \in X: \varrho_x(a) = U\}, \quad U \subset V_a$

(3) $\sigma(\underline{C}(a, U)) = \{x \in X: \varrho_x(a) \subset U\}$

(4) $\sigma(C(a, U)) = \{x \in X: \varrho_x(a) \supset U\}$

(5) $\sigma(\overline{C}(a, U)) = \{x \in X: \varrho_x(a) \cap U \neq \emptyset\}$

(6) $\sigma(t+t') = \sigma(t) \cap \sigma(t')$

(7) $\sigma(\underline{C}(t+t')) \supset \sigma(\underline{C}(t)) \cup \sigma(\underline{C}(t'))$

(8) $\sigma(C(t+t')) = \sigma(C(t)) \cup \sigma(C(t'))$

(9) $\sigma(\overline{C}(t+t')) = \sigma(\overline{C}(t)) \cup \sigma(\overline{C}(t'))$

(10) $\sigma(t \cdot t') = \sigma(t) \cap \sigma(t')$

(11) $\sigma(\underline{C}(t \cdot t')) = \sigma(\underline{C}(t)) \cap \sigma(\underline{C}(t'))$

(12) $\sigma(C(t \cdot t')) \subset \sigma(C(t)) \cap \sigma(C(t'))$

(13) $\sigma(\overline{C}(t \cdot t')) \subset \sigma(\overline{C}(t)) \cap \sigma(\overline{C}(t'))$

(14) $\sigma(\sim t) = X - \sigma(t)$

(15) $\sigma(\sim \underline{C}(t)) = X - \sigma(\underline{C}(t)) = \sigma(\overline{C}(\sim t))$

(16) $\sigma(\sim C(t)) = X - \sigma(C(t))$

(17) $\sigma(\sim \overline{C}(t)) = X - \sigma(\overline{C}(t)) = \sigma(\underline{C}(\sim t))$

(18) $\sigma(\underline{C}(\underline{C}(t))) = \sigma(\underline{C}(t))$

(19) $\sigma(\overline{C}(\underline{C}(t))) = \sigma(\overline{C}(t))$

(20) $\sigma(C(\underline{C}(t))) = \sigma(C(t))$

(21) $\sigma(\underline{C}(\overline{C}(t))) = \sigma(\overline{C}(t))$

(22) $\sigma(\overline{C}(\overline{C}(t))) = \sigma(\overline{C}(t))$

(23) $\sigma(C(\overline{C}(t))) = \sigma(\overline{C}(t))$

(24) $\sigma(\underline{C}(C(t))) \supset \sigma(C(t))$

(25) $\sigma(\overline{C}(C(t))) \supset \sigma(C(t))$

(26) $\sigma(C(C(t))) \supset \sigma(C(t))$

Dla uproszczenia pominęliśmy tu operacje $\rightarrow$, $\leftrightarrow$.

Obecnie zilustrujemy prostymi przykładami niektóre ciekawsze reguły określające semantykę.

Reguła 2 określa, że znaczeniem deskryptora (a, U), $U \subset V_a$ jest po prostu blok relacji $\underset{a}{\sim}$, którego obiekty mają jednakową informację U. Na przykład jeżeli $U = (\text{an, pl})$, to znaczeniem deskryptora (JĘZYK = (an, pl)) jest zbiór wszystkich osób znających jednocześnie angielski i polski i nie znających żadnych innych języków.

Reguła 3 określa, że znaczeniem wyrażenia $\underline{C}(a, U)$ jest zbiór tych wszystkich obiektów systemu, które są opisane każdym deskryptorem postaci (a, U') $U' \subset U$. Na przykład jeżeli $U = (\text{an, pl})$, to wyrażenie (JĘZYK = (an, pl)) opisuje zbiór osób mówiących tylko po polsku lub tylko po angielsku lub znających jednocześnie języki polski i angielski oraz nie znających żadnego innego języka.

Reguła 4 określa znaczenie wyrażenia $C(a, U)$ jako zbiór tych wszystkich obiektów systemu, które są opisane dowolnym deskryptorem postaci (a, U'), $U' \supset U$, np. jeśli $U = (\text{pl, an})$, to C(JĘZYK = (pl, an)) oznacza zbiór tych wszystkich osób, które znają jednocześnie język polski i angielski i być może jeszcze inny język, np. francuski.

Reguła 5 określa znaczenie wyrażenia $\bar{C}(a, U)$ jako zbiór tych wszystkich obiektów systemu, które są opisane dowolnym deskryptorem postaci (a, U'), $U' \cap U \neq 0$, np. gdy $U = (\text{an, pl})$, wówczas wyrażenie $\bar{C}$(JĘZYK = (an, pl)) oznacza zbiór wszystkich osób znających język polski lub angielski, bądź oba te języki, a więc do tego zbioru należą wszystkie osoby znające polski oraz wszystkie osoby znające język angielski oraz być może inne języki.

Widać stąd, że termy $\underline{C}(a, U)$, $C(a, U)$, $\bar{C}(a, U)$ można przedstawić tak, jak to pokazano poniżej:

(a) $\underline{C}(a, U) = \sum_{t \in T(a,U)_*} t$

(b) $C(a, U) = \sum_{t \in T(a,U)} t$

(c) $\bar{C}(a, U) = \sum_{t \in T(a,U)^*} t$

Zbiór $T_{(a,U)_*}$ jest zbiorem wszystkich termów prostych, w których występują deskryptory postaci (a, U'), gdzie $U' \subset U$; zbiór $T_{(a,U)}$ jest zbiorem wszystkich termów prostych, w których występują deskryptory postaci (a, U'), gdzie $U' \supset U$; zbiór $T_{(a,U)^*}$ jest zbiorem wszystkich termów prostych, w których występują deskryptory postaci (a, U'), takich, że $U' \cap U \neq \emptyset$.

Termy $\underline{C}(a, U)$, $C(a, U)$, $\bar{C}(a, U)$ można też przedstawić w postaci:

(d) $\underline{C}(a, U) = \sum_{U' \subset U} (a, U')$

(e) $C(a, U) = \sum_{U' \supset U} (a, U')$

(f) $\bar{C}(a, U) = \sum_{U' \cap U \neq \emptyset} (a, U')$

Widać stąd, że operatory $\underline{C}$, C, $\overline{C}$ stojące przed deskryptorami można wyeliminować, zastępując wyrażenia $\underline{C}(a, U)$, $C(a, U)$, $\overline{C}(a, U)$ przez sumy odpowiednich deskryptorów.

Zilustrujemy obecnie niektóre z pozostałych reguł. Jeżeli w regule (7) jako t przyjmiemy (JĘZYK = (an, pl)), jako t' zaś przyjmiemy (JĘZYK = (an, fr)), to $\underline{C}(t+t')$ będzie oznaczało zbiór tych wszystkich osób, które znają którykolwiek z języków: polski, angielski, francuski i jednocześnie nie znają żadnych innych języków. Do zbioru tego należą również osoby znające jednocześnie język angielski, polski i francuski. Natomiast do zbioru $\underline{C}(t)+\underline{C}(t')$ nie należą osoby znające jednocześnie język angielski, polski i francuski, gdyż zgodnie z definicją $\underline{C}(t)$ oznacza zbiór osób znających tylko język polski, tylko język angielski lub tylko jednocześnie język polski i angielski. Wyrażenie $\underline{C}(t')$ oznacza zbiór osób znających tylko język angielski, tylko język francuski bądź tylko angielski i francuski. A więc do tego zbioru nie należą osoby znające jednocześnie trzy języki: angielski, polski, francuski.

Podobnie w przypadku reguły (12). Jeżeli przyjmiemy taką samą interpretację termów t, t', to znaczeniem wyrażenia $C(t \cdot t')$ jest zbiór pusty, gdyż t i t' oznaczają dwa różne bloki relacji $\underset{a}{\sim}$, zaś $C(t)$ oznacza zbiór wszystkich osób znających jednocześnie język polski i angielski; $C(t')$ oznacza zbiór wszystkich osób znających jednocześnie język angielski i francuski, a więc w obu zbiorach $C(t)$ i $C(t')$ może występować zbiór osób znających jednocześnie języki angielski, polski i francuski. W konsekwencji iloczyn $C(t) \cdot C(t')$ nie jest pusty.

Regułę (13) można zilustrować podobnie jak poprzednie. Również w tym przypadku $\overline{C}(t \cdot t')$ oznacza zbiór pusty, gdyż t i t' oznaczają dwa różne bloki relacji $\underset{a}{\sim}$, natomiast $\overline{C}(t)$ jest zbiorem wszystkich osób znających język polski bądź język angielski lub też oba te języki, $\overline{C}(t')$ oznacza zbiór wszystkich osób znających język angielski lub język francuski, bądź oba te języki. W obu przypadkach osoby te mogą znać również inne języki oprócz wymienionych. W konsekwencji w obu zbiorach odpowiadających wyrażeniom $\overline{C}(t)$ oraz $\overline{C}(t')$ może się znaleźć zbiór osób znających jednocześnie język polski i angielski, a więc iloczyn $\overline{C}(t) \cdot \overline{C}(t')$ jest niepusty.

Jeżeli żaden atrybut nie występuje jednocześnie w termach t i t', to reguły (7), (12), (13) można zastąpić regułami:

(7′) $\sigma(\underline{C}(t+t')) = \sigma(\underline{C}(t)) \cup \sigma(\underline{C}(t'))$

(12′) $\sigma(C(t \cdot t')) = \sigma(C(t)) \cap \sigma(C(t'))$

(13′) $\sigma(\overline{C}(t \cdot t')) = \sigma(\overline{C}(t)) \cap \sigma(\overline{C}(t'))$

5.6. AKSJOMATY JĘZYKA WIELOWARTOŚCIOWEGO

Dla przekształcania wyrażeń języka wielowartościowego przyjmiemy aksjomaty algebry Boole'a oraz następujące aksjomaty specyficzne, charakteryzujące deskryptory wielowartościowe:

A1. $\underline{C}((a, U)+(a, U'))=\underline{C}(a, U)+\underline{C}(a, U')$

A2. $\underline{C}((a, U)\cdot(a, U'))=\underline{C}(a, U)\cdot\underline{C}(a, U')$

A3. $\sim\underline{C}(a, U)=\bar{C}(a, V_a-U)$

A4. $\bar{C}((a, U)+(a, U))=\bar{C}(a, U)+\bar{C}(a, U')$

A5. $\bar{C}((a, U)\cdot(a, U))=\bar{C}(a, U)\cdot\bar{C}(a, U')$

A6. $\sim\bar{C}(a, U)=\underline{C}(a, V_a-U)$

A7. $C((a, U)+(a, U))=C(a, U)+C(a, U')$

A8. $C((a, U)\cdot(a, U))=C(a, U)\cdot C(a, U')$

A9. $\sim C(a, U)=1-C(a, U)$

Aksjomat A1 jest oczywisty. Jeżeli np. U oznacza jednoczesną znajomość języka angielskiego i polskiego, U' zaś — francuskiego i niemieckiego, to $\underline{C}((a, U)+(a, U'))$ oznacza zbiór tych wszystkich osób, które znają tylko język polski bądź tylko język angielski, bądź też tylko język polski i angielski, oraz tych osób, które znają wyłącznie język francuski bądź język niemiecki, bądź też znają jednocześnie język francuski i niemiecki. Nie ma więc w tym zbiorze osób znających jednocześnie język polski i francuski, bądź angielski i niemiecki itd.

Z aksjomatu A2 wynika, że jeżeli $U\cap U'=\emptyset$, to $\underline{C}((a, U)\cdot(a, U'))$ oznacza również zbiór pusty. W przeciwnym przypadku, tj. gdy $U\cap U'\neq 0$, znaczeniem termu $\underline{C}((a, U)\cdot(a, U'))$ jest zbiór niepusty. Na przykład gdy U oznacza język polski i angielski, U' zaś angielski i francuski, wówczas $\underline{C}((a, U)\cdot(a, U'))$ oznacza zbiór osób znających tylko język angielski.

Aksjomat A3 charakteryzuje negację. Jeżeli np. rozpatrujemy osoby mogące znać języki angielski, francuski, polski oraz za U przyjmiemy polski i angielski, to wyrażenie $\sim\underline{C}(a, U)$ oznacza zbiór tych wszystkich osób, które znają jakieś języki poza polskim i angielskim, tj. w naszym przypadku francuski.

Aksjomat A4 nie wymaga wyjaśnień.

W aksjomacie A5, jeżeli $U\cap U'=\emptyset$, to w przeciwieństwie do aksjomatu A3, $\bar{C}((a, U)\cdot(a, U'))$ może oznaczać zbiór niepusty. Na przykład jeżeli U oznacza język polski i angielski, U' zaś francuski i niemiecki, to wyrażenie $\bar{C}((a, U)\cdot(a, U'))$ oznacza zbiór tych wszystkich osób, które jednocześnie znają jakieś języki z U oraz z U', np. do zbioru tego należeć będą osoby znające jednocześnie język francuski i angielski.

Na podstawie aksjomatu A6, dualnego do aksjomatu A3, zbiór osób, np. nie znających co najmniej języka polskiego lub angielskiego jest identyczny ze zbiorem osób znających wyłącznie jakiekolwiek inne języki, np. francuski.

Ostatnia grupa aksjomatów dotyczy operacji C. Aksjomat A7 nie wymaga omówienia. Aksjomat A8 mówi, że jeżeli np. U oznacza język polski i angielski, U' zaś język francuski i niemiecki, to wyrażenie $C((a, U)\cdot(a, U'))$ oznacza zbiór tych wszystkich osób, które znają co najmniej jednocześnie język polski, angielski, francuski i niemiecki.

Wreszcie ostatni aksjomat A9, charakteryzujący negację, stwierdza, że np. zbiór osób nie znających co najmniej jednocześnie języka angielskiego i polskiego jest identyczny ze zbiorem osób, z których żadna nie zna jednocześnie języka angielskiego i polskiego.

Dla lepszego scharakteryzowania podanych powyżej aksjomatów, podamy kilka własności wynikających bezpośrednio z własności wyrażeń $\bar{C}(a, U)$, $C(a, U)$, $\underline{C}(a, U)$.

(1) $\underline{C}(a, U)+\underline{C}(a, U')\subset\underline{C}(a, U\cup U')$

(2) $\bar{C}(a, U)+\bar{C}(a, U')=\bar{C}(a, U\cup U')$

(3) $C(a, U)+C(a, U')\subset C(a, U\cup U')$

(4) $\underline{C}(a, U)\cdot\underline{C}(a, U')=\underline{C}(a, U\cap U')$

(5) $\bar{C}(a, U)\cdot\bar{C}(a, U')\subset\bar{C}(a, U\cap U')$

(6) $C(a, U)\cdot C(a, U')=C(a, U\cap U')$

Z praktycznego punktu widzenia nie wszystkie termy języka, w których występują operacje $\underline{C}$, C, $\bar{C}$, są interesujące. Sensowne praktycznie wydają się być jedynie termy, w których operatory $\underline{C}$, C, $\bar{C}$ występują jedynie bezpośrednio przy deskryptorach, jak np.

$\bar{C}(a, U)+\bar{C}(b, V)$

$C(a, U)\cdot C(a, U')$

$\underline{C}(a, U)+\underline{C}(b, V)\cdot C(a, U')$,

W termach tego rodzaju operatory $\bar{C}$, C, $\underline{C}$ można wyeliminować w prosty sposób na podstawie reguł (3), (4), (5) podanych w p. 5.5.

Przykład 5.3

Rozpatrzmy term

$t=(\bar{C}(a, U)+\underline{C}(a, V))\cdot C(a, W)$

w którym:

$U=\{u, v\}$

$V=\{v, w\}$

$W=\{u, w\}$

Na podstawie definicji semantyki — można napisać

$\bar{C}(a, U)=(a, u)+(a, v)+(a, u, v)+(a, v, w)+(a, u, w)+(a, u, v, w)$

$\underline{C}(a, V)=(a, v)+(a, w)+(a, v, w)$

$C(a, W)=(a, u, w)+(a, u, w, v)$

Po wymnożeniu otrzymamy

$t=(a, u, w)+(a, u, v, w)$ ■

Przypominamy, że iloczyn dwu deskryptorów postaci (a, v), (a, v'), $v\neq v'$ jest równy zeru, dla dowolnego a, v i v'.

6. PRZYBLIŻONE SYSTEMY INFORMACYJNE

6.1. WPROWADZENIE

W dotychczasowych rozważaniach przyjmowaliśmy, że zarówno w systemach jednowartościowych, jak i wielowartościowych wartość każdego atrybutu jest znana i jednoznacznie określona. Istnieją jednakże sytuacje, w których nie można przyjąć takiego założenia, np. w kartotekach kryminalnych. Często brak jest dostatecznych danych o przestępcach figurujących w takiej kartotece. Możemy mieć o nich tylko dane częściowe. Na przykład możemy nie znać dokładnie wieku przestępcy, jego koloru włosów czy też innych cech. Podobna sytuacjach może istnieć w bibliotece starodruków. Na skutek zniszczenia książki możemy nie znać jej autora, tytułu, daty wydania czy też wydawcy. Również w przypadku opisu stanu pacjenta w kartotece medycznej możemy mieć trudności w zebraniu wszystkich potrzebnych danych. W opisie znalezisk archeologicznych sytuacja jest również podobna.

Brak odpowiednich danych nie pozwala w pełni określić szukanych obiektów. W sytuacji takiej naturalne wydaje się założenie, że odpowiedzią na pytanie powinien być zbiór obiektów taki, który na pewno zawiera interesujące nas obiekty, chociaż nie potrafimy ich dokładnie w tym zbiorze wyróżnić — a więc zbiór większy od poszukiwanego. Gdy pytamy np. o przęstępców, którzy mogli popełnić określone przestępstwo, a brak wszystkich danych nie pozwala na ich pełną identyfikację, chcielibyśmy, aby wśród osób, które można określić na podstawie posiadanych danych jako potencjalnych sprawców określonego czynu kryminalnego, na pewno znalazły się osoby, które ten czyn popełniły. Co więcej, chcielibyśmy, aby w przypadku gdy otrzymamy nowe dane, np. w wyniku śledztwa, zbiór podejrzanych się zmniejszył, jednakże w taki sposób, aby nadal na pewno się w nim znajdowali potencjalni sprawcy przestępstwa.

Dla opisu tego typu sytuacji wprowadzimy pojęcie przybliżonego systemu informacyjnego o obiektach. Przyjmiemy dla uproszczenia, że atrybuty systemu są jednowartościowe, jednakże dopuszczamy możliwość, że wartość atrybutu dla jakiegoś obiektu jest nieznana. Ściślej — dopuścimy, że atrybut a przyjmuje jedną z wartości podzbioru $U \subset V_a$, nie wiemy tylko którą z nich. Na przykład, jeśli jako atrybut z niepełną informacją przyjmiemy KOLOR, to deskryptor (KOLOR = niebieski, czerwony) oznacza, że w rzeczywistości atrybut ten przyjmuje dla konkretnego obiektu tylko wartość niebieski bądź czerwony, na pewno żaden inny i że w trakcie jakichkolwiek badań mających na celu wyjaśnienie, jaki kolor ma faktycznie badany przedmiot, nie może się okazać, że jest on zielony czy biały, może być tylko niebieski lub czerwony.

Badaniem takich systemów informacyjnych zajmowali się między innymi Jaegermann [22], [23], Lipski [37], [38], Sendova [102], tutaj jednak przyjmiemy inny punkt widzenia niż zaprezentowany we wspomnianych pracach.

6.2. DEFINICJE PRZYBLIŻONEGO SYSTEMU INFORMACYJNEGO

Definicja

Przybliżonym systemem informacyjnym będziemy nazywali czwórkę

$$S = \langle X, A, V, \varrho \rangle$$

gdzie: X — zbiór obiektów, A — zbiór atrybutów, $V = \bigcup_{a \in A} V_a$ — zbiór wartości atrybutów, $\varrho\colon X \times A \to \mathscr{P}(V)$ — funkcja całkowita taka, że $\varrho(x, a) \subset V_a$, dla każdego $x \in X$, $a \in A$. Przez ϱ_x: $A \to \mathscr{P}(V)$ będziemy rozumieli funkcję taką, że $\varrho_x(a) = \varrho(x, a)$ dla każdego $a \in A$, $x \in X$. ■

Podobnie jak w przypadku systemów wielowartościowych obiekty x, $y \in X$ są *równoważne w systemie S ze względu na atrybut a*, gdy $\varrho_x(a) = \varrho_y(a)$ oraz są *równoważne ze względu na każdy atrybut w A*, gdy $\varrho_x = \varrho_y$. Oczywiście relacje $\underset{a}{\sim}$, $\underset{S}{\sim}$ są również tutaj relacjami równoważności.

Parę (a, U), gdzie $a \in A$, $U \subset V_a$ oraz U jest zbiorem co najmniej dwuelementowym, będziemy nazywać *deskryptorem przybliżonym*; jeżeli U jest zbiorem jednoelementowym, deskryptor (a, U) nazwiemy — *dokładnym.*

6.3. PRZYKŁAD SYSTEMU PRZYBLIŻONEGO

Rozważmy przybliżony system informacyjny z jednym atrybutem KOLOR, mogącym przyjmować wartości ze zbioru trójelementowego {niebieski, czerwony, zielony}. Możemy mieć w naszym systemie deskryptory dokładne, jak na przykład (KOLOR = niebieski), bądź przybliżone, jak np. (KOLOR = niebieski, zielony). Dla uproszczenia, gdy nie będzie to groziło nieporozumieniem, będziemy pomijać nawiasy w podzbiorach atrybutów, tak jak to uczyniono w powyższym przykładzie. Tabelka tego systemu jest następująca:

X	KOLOR
x_1	n
x_2	n, c
x_3	n
x_4	n, c
x_5	n, c, z
x_6	z
x_7	c, z
x_8	c, z
x_9	n, c, z
x_{10}	z
x_{11}	n

Litery n, c, z oznaczają odpowiednio kolory niebieski, czerwony, zielony.

W systemie tym jest pięć zbiorów elementarnych:

$E_n = \{x_1, x_3, x_{11}\}$

$E_z = \{x_6, x_{10}\}$

$E_{n,c} = \{x_2, x_4\}$

$E_{c,z} = \{x_7, x_8\}$

$E_{n,c,z} = \{x_5, x_9\}$

W zbiorze elementarnym E_n znajdują się wyłącznie przedmioty niebieskie, w zbiorze E_z — wyłącznie przedmioty zielone. O zbiorze $E_{n,c}$ wiemy tylko tyle, że nie ma w nim przedmiotów zielonych, może on natomiast zawierać tylko przedmioty niebieskie bądź tylko czerwone, bądź też zarówno niebieskie, jak i czerwone. Podobnie zbiór $E_{c,z}$ na pewno nie zawiera przedmiotów niebieskich, natomiast mogą się w nim znajdować przedmioty zarówno czerwone, jak i zielone, bądź tylko w jednym z powyższych kolorów. W zbiorze $E_{n,c,z}$ mogą znajdować się przedmioty o dowolnym kolorze. Może on więc zawierać tylko przedmioty niebieskie, czerwone bądź zielone — bądź też przedmioty o dowolnych dwu kolorach, np. niebieskie i czerwone, albo przedmioty o wszystkich kolorach.

Nie wiemy więc dokładnie, jakiego koloru przedmioty znajdują się w zbiorach elementarnych $E_{n,c}$, $E_{c,z}$, $E_{n,c,z}$, wiemy tylko jakich kolorów ewentualnie w nich nie ma. O przedmiotach w tych zbiorach mamy więc jedynie informację przybliżoną. Natomiast o przedmiotach należących do zbiorów elementarnych E_n, E_z mamy informację dokładną.

Gdybyśmy w naszym systemie chcieli utworzyć zbiór przedmiotów, które są w rzeczywistości np. niebieskie, okazałoby się to niemożliwe. Wiemy bowiem, że przedmioty niebieskie znajdują się w zbiorze E_n i wiemy, że na pewno nie ma ich w zbiorze E_z, natomiast w pozostałych zbiorach elementarnych mogą one się znajdować, bądź nie — nie jest to wykluczone, natomiast pewności co do tego nie mamy. W konsekwencji zbioru przedmiotów niebieskich nie możemy w naszym systemie utworzyć. Możemy podać jedynie dolne i górne przybliżenie tego zbioru w naszym systemie, tj. podać największy zbiór opisywalny w tym systemie zawarty w zbiorze przedmiotów niebieskich, tj. zbiór elementarny E_n, bądź też najmniejszy zbiór opisywalny zawierający być może przedmioty niebieskie, tj. $E_n \cup E_{n,z} \cup E_{n,c,z}$. Nie wiemy na pewno, czy w zbiorach $E_{n,z}$ oraz $E_{n,c,z}$ znajdują się przedmioty niebieskie, ale nie możemy tego faktu wykluczyć, dlatego zbiory te muszą się znaleźć w górnym przybliżeniu zbioru przedmiotów niebieskich.

Dolnym przybliżeniem zbioru przedmiotów niebieskich lub czerwonych będzie w naszym systemie zbiór $E_{n,c} \cup E_n$, górnym zaś zbiór $E_{n,c} \cup E_n \cup E_{c,z} \cup E_{n,c,z}$.

Dokładna analiza tego przykładu pozwoli nam zrozumieć semantykę języka systemów przybliżonych. ■

6.4. SKŁADNIA JĘZYKA PRZYBLIŻONEGO SYSTEMU INFORMACYJNEGO

Z rozważanego w poprzednim punkcie przykładu wynika, że w przypadku przybliżonych systemów informacyjnych można zadawać dwa całkowicie różne rodzaje pytań, pytania dotyczące rzeczywistości i pytania dotyczące informacji znajdującej się w systemie. Na pierwszy rodzaj pytań nie możemy uzyskać dokładnej odpowiedzi, możemy jedynie otrzymać od systemu odpowiedzi przybliżone (od dołu albo od góry) — natomiast na drugi rodzaj pytań możemy uzyskiwać odpowiedzi dokładne.

Język nasz zbudujemy w ten sposób, aby można było w nim zadawać oba rodzaje pytań.

Niech $S = \langle X, A, V, \varrho \rangle$ będzie przybliżonym systemem informacyjnym. Alfabet języka L_S będzie się składał z następujących symboli:

(1) stałych 0, 1,

(2) deskryptorów (a, U), $U \subset V_a$, $a \in A$,

(3) symboli operacji $\sim$, $+$, $\cdot$, $\rightarrow$, $\leftrightarrow$,

(4) symboli operacji $\underline{C}$, $\bar{C}$.

Termem w języku L_S będziemy nazywali wyrażenie zdefiniowane indukcyjnie w następujący sposób:

(1) Stałe 0, 1 są termami.

(2) Każdy deskryptor (a, U), $U \subset V_a$, $a \in A$ jest termem.

(3) Jeżeli t, t' są termami, to również termami są wyrażenia $(t+t')$, $(t \cdot t')$, $(t \rightarrow t')$, $(t \leftrightarrow t')$, $\sim t$.

(4) Jeżeli t jest termem, to termami są również $\underline{C}(t)$, $\bar{C}(t)$.

Termami są więc np. wyrażenia $\underline{C}(0)$, $\bar{C}(1)$, $\underline{C}(a, U)$, $\bar{C}(a, U)+(b, U')$, $\underline{C}(\bar{C}(a, U)+ +\underline{C}(b, U''))$ itd.

Jeżeli w powyższej definicji w punkcie (2) zamiast dowolnych deskryptorów dopuścimy tylko deskryptory dokładne, tj. deskryptory postaci (a, v), gdzie $v \in V_a$, $a \in A$ — oraz opuścimy punkt (4) tej definicji, to tak otrzymany język będziemy nazywać *językiem dokładnym* systemu S, w przeciwieństwie do języka z operatorem $\underline{C}$, $\bar{C}$ oraz deskryptorami przybliżonymi — który będziemy nazywać *językiem przybliżonym* systemu S. Oczywiście język dokładny systemu S jest podjęzykiem (w sensie podzbioru termów języka L_S) języka przybliżonego systemu S.

Przykładami termów w języku dokładnym systemu S mogą być np. wyrażenia $((a, V)+\sim(b, u))\cdot(c, w)$, $(a, v)\cdot(b, u)$ itd. lub też (KOLOR = niebieski)+(KOLOR = czerwony) itd.

Język dokładny, jak to już wspominaliśmy, służy do zadawania pytań o rzeczywistości, język przybliżony natomiast do pytania o informacje znajdujące się w systemie. Sprawy te wyjaśnimy dokładnie, dyskutując semantykę języka przybliżonego.

6.5. SEMANTYKA JĘZYKA PRZYBLIŻONEGO SYSTEMU INFORMACYJNEGO

Zgodnie z tym, co powiedzieliśmy poprzednio, semantyki języka przybliżonego systemu nie można zdefiniować, gdyż znaczeniem wyrażenia języka dokładnego mogą być zbiory w rozważanym systemie nieopisywalne (nie będące sumą zbiorów elementarnych systemu).

Semantyka języka przybliżonego systemu informacyjnego $S = \langle X, A, V, \varrho \rangle$, tj. funkcja σ, musi spełniać następujące warunki:

(1) $\sigma(0) = \sigma(\underline{C}(0)) = \sigma(\overline{C}(0)) = \emptyset$

(2) $\sigma(1) = \sigma(\underline{C}(1)) = \sigma(\overline{C}(1)) = X$

(3) $\sigma(a, U) = \{x \in X: \varrho_x(a) = U, U \subset V_a\}$, $a \in A$

(4) $\sigma(\underline{C}(a, U)) = \{x \in X: \varrho_x(a) \subset U\}$

(5) $\sigma(\overline{C}(a, U)) = \{x \in X: \varrho_x(a) \cap U \neq \emptyset\}$

(6) $\sigma(t+t') = \sigma(t) \cup \sigma(t')$

(7) $\sigma(\underline{C}(t+t')) \supset \sigma(\underline{C}(t)) \cup \sigma(\underline{C}(t'))$

(8) $\sigma(\overline{C}(t+t')) = \sigma(\overline{C}(t)) \cup \sigma(\overline{C}(t'))$

(9) $\sigma(t \cdot t') = \sigma(t) \cap \sigma(t')$

(10) $\sigma(\underline{C}(t \cdot t')) = \sigma(\underline{C}(t)) \cap \sigma(\underline{C}(t'))$

(11) $\sigma(\overline{C}(t \cdot t')) \subset \sigma(\overline{C}(t)) \cap \sigma(\overline{C}(t'))$

(12) $\sigma(\sim t) = X - \sigma(t)$

(13) $\sigma(\sim \underline{C}(t)) = \sigma(\overline{C}(\sim t))$

(14) $\sigma(\sim \overline{C}(t)) = \sigma(\underline{C}(\sim t))$

(15) $\sigma(\overline{C}(\overline{C}(t))) = \sigma(\overline{C}(t))$

(16) $\sigma(\underline{C}(\overline{C}(t))) = \sigma(\overline{C}(t))$

(17) $\sigma(\underline{C}(\underline{C}(t))) = \sigma(\underline{C}(t))$

(18) $\sigma(\overline{C}(\underline{C}(t))) = \sigma(\underline{C}(t))$

Dla uproszczenia pominęliśmy tu operacje $\rightarrow$, $\leftrightarrow$.

Jako aksjomaty przyjmiemy aksjomaty algebry Boole'a oraz następujące aksjomaty charakteryzujące deskryptory:

A1. $\underline{C}((a, U)+(a, U')) = \underline{C}(a, U \cup U')$

A2. $\underline{C}((a, U) \cdot (a, U')) = \underline{C}(a, U \cap U')$

A3. $\sim \underline{C}(a, U) = \bar{C}(a, V_a - U)$

A4. $\bar{C}((a, U)+(a, U')) = \bar{C}(a, U \cup U')$

A5. $\bar{C}((a, U) \cdot (a, U')) = \bar{C}(a, U \cap U')$

A6. $\sim \bar{C}(a, U) = \underline{C}(a, V_a - U)$

Powyższe aksjomaty zilustrujemy prostymi przykładami.

Z aksjomatu A1 wynika, że wyrażenie $\underline{C}$((KOLOR = niebieski, czerwony) + + (KOLOR = niebieski, zielony)) można zastąpić wyrażeniem $\underline{C}$(KOLOR = niebieski, czerwony, zielony). Na podstawie aksjomatu A2 powyższe wyrażenie można zastąpić wyrażeniem $\underline{C}$(KOLOR = niebieski). Podobnie dla wyrażeń zawierających operator $\bar{C}$.

Z aksjomatu A3 wynika, że wyrażenie $\sim \bar{C}$(KOLOR = niebieski, czerwony) można zastąpić wyrażeniem $\bar{C}$(KOLOR = zielony), tzn. zbiór obiektów, które nie są na pewno ani niebieskie, ani czerwone, jest równy zbiorowi obiektów, które są być może zielone.

Na podstawie aksjomatu A6 wyrażenie $\sim \bar{C}$(KOLOR = niebieski, czerwony) można zastąpić wyrażeniem $\underline{C}$(KOLOR = zielony), a więc zbiór obiektów, które nie są być może niebieskie bądź czerwone jest równy zbiorowi obiektów, które są na pewno zielone.

Mówiąc niezbyt ściśle, aksjomaty A1, A4 określają kiedy elementarne własności można zastąpić ich sumą teoriomnogościową; aksjomaty A2, A5 określają, kiedy koniunkcje własności można zastąpić ich iloczynem teoriomnogościowym, aksjomaty zaś A3, A6 wyjaśniają znaczenie negacji deskryptora.

Zilustrujemy obecnie ważniejsze z podanych reguł semantycznych przykładami, posługując się systemem rozważanym w poprzednim punkcie.

Reguła (3) mówi, że znaczeniem deskryptora (a, U) jest blok relacji $\underset{a}{\sim}$ taki, którego wszystkie elementy x mają informacje $\varrho_x(a) = U$. Na przykład deskryptor (KOLOR = czerwony) oznacza zbiór elementarny E_c, tj. zbiór wszystkich tych obiektów systemu, o których wiemy, że mają one kolor czerwony. Deskryptor (KOLOR = czerwony, niebieski) oznacza zbiór elementarny $E_{c,n}$, tj. zbiór tych wszystkich obiektów systemu, o których wiemy, że nie mają koloru zielonego, mają natomiast kolor czerwony lub niebieski, ale nie wiemy który z nich. Mówimy tu o znaczeniu w języku dokładnym systemu. W języku dokładnym systemu deskryptor (KOLOR = czerwony) — oznacza zbiór wszystkich obiektów systemu, które są w rzeczywistości czerwone, ale jak to już pisaliśmy, obiektów tych nie znamy i w systemie nie można ich zdefiniować.

Reguła (4) definiuje znaczenie wyrażenia $\underline{C}(a, U)$ jako sumę tych wszystkich bloków relacji $\underset{a}{\sim}$, których deskryptory mają postać (a, U'), $U' \subset U$. Na przykład znaczeniem deskryptora (KOLOR = czerwony, niebieski) jest zbiór $E_{c,n} \cup E_c \cup E_n$.

Reguła (5) określa znaczenie deskryptora jako sumę wszystkich bloków relacji $\underset{a}{\sim}$, których deskryptory mają postać (a, U'), $U' \cap U \neq \emptyset$. Na przykład deskryptor (KOLOR = czerwony, niebieski) oznacza zbiór $E_{c,n} \cup E_c \cup E_n \cup E_{c,z} \cup E_{n,z} \cup \cap E_{c,z,n}$.

Mówiąc niezbyt ściśle, deskryptor (a, U) oznacza zbiór tych wszystkich obiektów systemu, o których wiemy, że ma on jedną z własności należących do U, ale nie wiemy którą. Zastosowanie operatora $\underline{C}$ do deskryptora (a, U) daje nam największy zbiór wszystkich obiektów systemu, o których wiemy, że mają one którąś z własności należących do U. Natomiast zastosowanie operatora $\overline{C}$ do deskryptora (a, U) daje najmniejszy zbiór wszystkich obiektów, o których wiemy, że posiadanie przez nich którejś z własności należących do U jest nie wykluczone.

Reguła (6) jest oczywista. Regułę (7) można zilustrować następującym przykładem: niech $t = (\text{KOLOR} = \text{czerwony})$ oraz $t' = (\text{KOLOR} = \text{niebieski})$. Wtedy $\underline{C}\,(t+t')$ na podstawie aksjomatu A1 można zastąpić wyrażeniem $\underline{C}(\text{KOLOR} = \text{czerwony, niebieski})$. Jego znaczeniem jest zbiór $E_c \cup E_n \cup E_{n,z}$. Natomiast znaczeniem wyrażenia $\underline{C}(\text{KOLOR} = \text{czerwony}) + \underline{C}(\text{KOLOR} = \text{niebieski})$ jest zbiór $E_c \cup E_n$. Równość więc nie zachodzi.

Reguły (8), (9), (10) nie wymagają wyjaśnienia. Regułę (11) zilustrujemy następującym przykładem. Niech $t = \overline{C}(\text{KOLOR} = \text{czerwony, zielony})$ oraz $t' = \overline{C}(\text{KOLOR} = \text{zielony, niebieski})$. Wyrażenie $\overline{C}(t \cdot t')$ na podstawie aksjomatu A5 można zastąpić wyrażeniem $\overline{C}(\text{KOLOR} = \text{zielony})$. Jego znaczeniem jest zbiór $E_z \cup E_{z,c} \cup E_{z,n} \cup E_{z,n,c}$. Natomiast znaczeniem wyrażenia $\overline{C}(t)$ jest zbiór $E_c \cup E_z \cup E_{c,n} \cup E_{n,z} \cup E_{c,n,z}$. Znaczeniem wyrażenia $\overline{C}(t')$ jest zbiór $E_z \cup E_n \cup E_{z,n} \cup E_{z,c} \cup E_{n,c} \cup E_{z,n,c}$. Iloczynem tych zbiorów jest $E_z \cup E_{z,n} \cup E_{z,c} \cup E_{n,z} \cup E_{c,z,n}$, a więc zawieranie zgodnie z regułą (11) zachodzi.

Reguła (12) również jest oczywista, natomiast reguły (13) i (14) wymagają pewnych wyjaśnień. Wiążą one operatory $\underline{C}$, $\overline{C}$ oraz negację. Wraz z aksjomatami A3 i A6 oraz prawami de Morgana wyjaśniają one, jak należy obliczać wartość termów zawierających negację oraz operatory $\underline{C}$ i $\overline{C}$.

6.6. OBLICZANIE WARTOŚCI TERMÓW W JĘZYKU PRZYBLIŻONYM

Podane reguły semantyczne oraz aksjomaty nie pozwalają obliczyć wartości termów języka przybliżonego, gdyż w regułach (7) i (11) zamiast równości występuje zawieranie. Jednak w przypadku, gdy w termach t, t' nie występują deskryptory z jednakowymi atrybutami, w regułach (7), (11) znak zawierania można zastąpić równością. Otrzymany wtedy zamiast reguł (7) i (11) reguły

$$(7')\quad \sigma(\underline{C}(t+t')) = \sigma(\underline{C}(t)) \cup \sigma(\underline{C}(t'))$$

$$(11')\quad \sigma(\overline{C}(t \cdot t')) = \sigma(\overline{C}(t)) \cup \sigma(\overline{C}(t'))$$

Przy tych założeniach można już obliczać wartości dowolnego termu w języku przybliżonym systemu.

Proces obliczania wartości termu polega na wyeliminowaniu z obliczanego termu negacji oraz symboli operacji $\underline{C}$, $\overline{C}$, a następnie sprowadzenia tak otrzymanego wyrażenia do postaci normalnej. Z postaci normalnej można już obliczyć znaczenie termu, tak samo jak w przypadku systemów jednowartościowych dokładnych.

Zwróćmy uwagę, że po wyeliminowaniu z termu symboli negacji oraz operacji $\underline{C}$, $\overline{C}$ mogą w nim znajdować się deskryptory — przybliżone. Znaczenie tych deskryptorów można obliczyć bezpośrednio z definicji i na tak otrzymanych zbiorach wykonać działania teoriomnogościowe wyznaczone przez uproszczony term.

Przykład 6.1

Dla zilustrowania powyższych wywodów rozważmy bardzo prosty przykład systemu przybliżonego z dwoma tylko atrybutami, KOLOR i WIELKOŚĆ, którego obiekty są charakteryzowane z przybliżeniem za pomocą tych dwu atrybutów. Przyjmiemy, że atrybut KOLOR może przyjmować tylko trzy wartości: niebieski, zielony, czerwony, atrybut WIELKOŚĆ zaś przyjmuje wartości: mały, średni, duży.

Dla uproszczenia nie będziemy podawać całej tabelki systemu, a jedynie jego niepuste zbiory elementarne. Przyjmiemy, że atrybut KOLOR generuje następujące, niepuste bloki B_z, B_c, $B_{n,c}$, $B_{n,z}$, $B_{n,z,c}$, atrybut WIELKOŚĆ zaś generuje niepuste bloki B_m, B_d, $B_{m,s}$, $B_{s,d}$. Wskaźniki u dołu są pierwszymi literami odpowiednich wartości atrybutów. System nasz ma więc następujące zbiory elementarne:

$$E_{z,m} = B_z \cap B_m$$

$$E_{z,d} = B_z \cap B_d$$

$$E_{z,m,s} = B_z \cap B_{m,s}$$

$$E_{z,s,d} = B_z \cap B_{s,d}$$

$$E_{c,m} = B_c \cap B_m$$

$$E_{c,d} = B_c \cap B_d$$

$$E_{c,m,s} = B_c \cap B_{m,s}$$

$$E_{c,s,d} = B_c \cap B_{s,d}$$

$$E_{n,c,m} = B_{n,c} \cap B_m$$

$$E_{n,c,d} = B_{n,c} \cap B_d$$

$$E_{n,c,m,s} = B_{n,c} \cap B_{m,s}$$

$$E_{n,c,s,d} = B_{n,c} \cap B_{s,d}$$

$$E_{n,z,m} = B_{n,z} \cap B_m$$

$$E_{n,z,d} = B_{n,z} \cap B_d$$

$$E_{n,z,m,s} = B_{n,z} \cap B_{m,s}$$

$$E_{n,z,s,d} = B_{n,z} \cap B_{s,d}$$

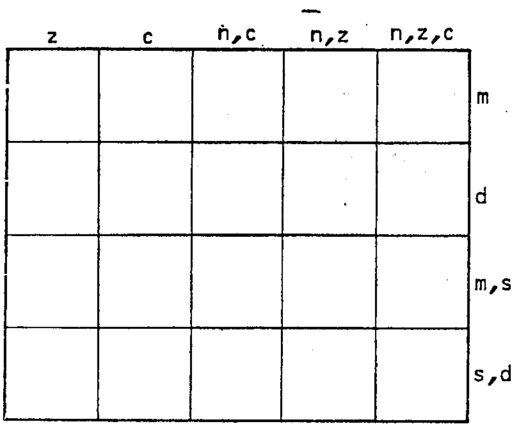
$$E_{n,z,c,m} = B_{n,z,c} \cap B_m$$

$$E_{n,z,c,d} = B_{n,z,c} \cap B_d$$

$$E_{n,z,c,m,s} = B_{n,z,c} \cap B_{m,s}$$

$$E_{n,z,c,s,d} = B_{n,z,c} \cap B_{s,d}$$

Podział obiektów na zbiory elementarne pokazano na rys. 6.1.



Rys. 6.1. Podział obiektów na zbiory elementarne

Wprowadzimy dla uproszczenia następujące oznaczenia:

$p =$ (KOLOR = niebieski, czerwony)

$q =$ (KOLOR = czerwony)

$r =$ (WIELKOŚĆ = mały, duży)

Obliczymy wartości kilku termów w języku przybliżonym zdefiniowanego systemu przybliżonego, posługując się tylko termami zbudowanymi z deskryptorów p, q, r.

Obliczymy najpierw wartość wyrażenia

(1) $\sim [(\underline{C}(\overline{C}(p) + \underline{C}(r))]$

Ponieważ p i r nie dotyczą tych samych atrybutów, stosując do wyrażenia (1) regułę (7′) otrzymamy

(2) $\sim [\underline{C}(\overline{C}(p)) + \underline{C}(\overline{C}(r))]$

Na podstawie reguł (16) oraz (17) możemy napisać

(3) $\sim [\overline{C}(p) + \underline{C}(r)]$

skąd stosując prawa de Morgana otrzymamy

(4) $\sim [(\overline{C}(p)) \cdot \sim (\underline{C}(r))$

Stosując do wyrażenia (4) aksjomaty A3 i A6, otrzymamy

(5) $\underline{C}(\sim p) \cdot \overline{C}(\sim r)$

co możemy zapisać jako

(6) $\underline{C}$(KOLOR = zielony) · $\overline{C}$(WIELKOŚĆ = średnia)

Stosując do wyrażenia (6) reguły (4) i (5), otrzymamy

(7) (KOLOR = zielony) · [(WIELKOŚĆ = małe, średnie) +
+ (WIELKOŚĆ = średnie, duże)]

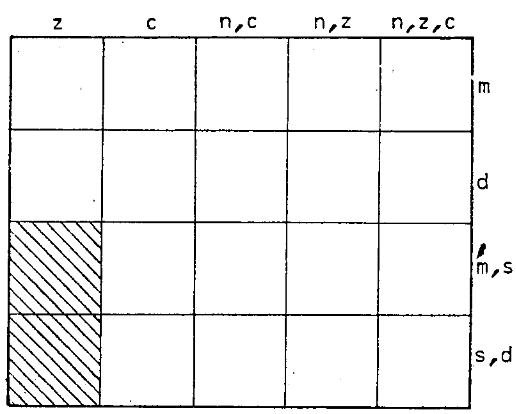
co po wymnożeniu daje postać normalną

(8) (KOLOR = zielony) · (WIELKOŚĆ = małe, średnie) +
+ (KOLOR = zielony) · (WIELKOŚĆ = średnie, duże)

W rezultacie znaczeniem wyrażenia (8) jest suma zbiorów elementarnych

$E_{z,m,s} \cup E_{z,s,d}$

Odpowiedź tę przedstawiono graficznie na rys. 6.2.



Rys. 6.2. Przykład odpowiedzi

W rezultacie odpowiedzią na nasze pytanie jest zbiór przedmiotów, o których wiemy, że są one zielone, natomiast nie znamy dokładnie ich wielkości. Informacja o nich posiadana w systemie, wyjaśnia, że są to przedmioty opisane jako małe lub średnie bądź jko średnie lub duże. Należy zauważyć, że suma przedmiotów, o których wiemy, że są małe lub średnie bądź średnie lub duże, nie jest wcale zbiorem przedmiotów, które są średnie, małe lub duże. Ten ostatni zbiór zawiera przedmioty, o których wielkości nie można nic powiedzieć na podstawie informacji posiadanych w systemie, natomiast zbiór przedmiotów otrzymanych jako odpowiedzi składa się z przedmiotów, o wielkości których mamy pewne informacje, a mianowicie o pewnych z nich można powiedzieć, że nie są małe, o innych zaś, że nie są duże.

Rozważmy w skrócie jeszcze jeden następujący term:

$\underline{C}(\sim \underline{C}(p)+\overline{C}(r)\cdot q)$

Przekształcając kolejno powyższy term otrzymamy

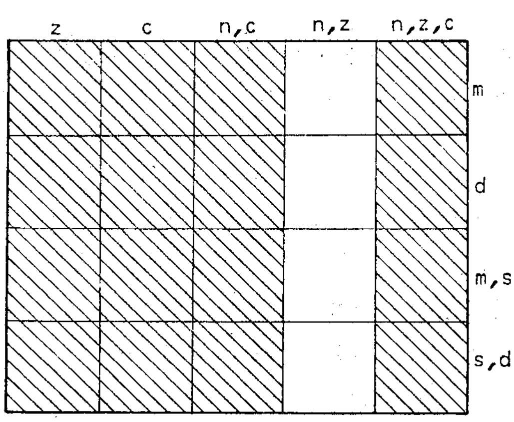
$\underline{C}(\overline{C}(\sim p)+\overline{C}(r)\cdot q) = \underline{C}(\overline{C}(\sim p))+\overline{C}(\overline{C}(r)\cdot q) = \overline{C}(\sim p)+\overline{C}(r)\cdot \overline{C}(q)$

Podstawiając za p, q, r odpowiadające im deskryptory, otrzymamy:

$\bar{C}$(KOLOR = zielony) + $\bar{C}$(WIELKOŚĆ = mały, duży) · $\bar{C}$(KOLOR =
= czerwony)

po przekształceniach otrzymamy postać normalną

(KOLOR = czerwony) · (WIELKOŚĆ = mały) +
(KOLOR = czerwony) · (WIELKOŚĆ = duży) +
(KOLOR = czerwony) · (WIELKOŚĆ = mały, średni) +
(KOLOR = czerwony) · (WIELKOŚĆ = średni, duży) +
(KOLOR = zielony) · (WIELKOŚĆ = mały) +
(KOLOR = zielony) · (WIELKOŚĆ = duży) +
(KOLOR = zielony) · (WIELKOŚĆ = mały, średni) +
(KOLOR = zielony) · (WIELKOŚĆ = średni, duży) +
(KOLOR = niebieski, czerwony) · (WIELKOŚĆ = mały) +
(KOLOR = niebieski, czerwony) · (WIELKOŚĆ = duży) +
(KOLOR = niebieski, czerwony) · (WIELKOŚĆ = mały, średni) +
(KOLOR = niebieski, czerwony) · WIELKOŚĆ = średni, duży) +
(KOLOR = niebieski, zielony, czerwony) · (WIELKOŚĆ = mały) +
(KOLOR = niebieski, zielony, czerwony) · (WIELKOŚĆ = duży) +
(KOLOR = niebieski, zielony, czerwony) · (WIELKOŚĆ = mały, średni) +
(KOLOR = niebieski, zielony, czerwony) · (WIELKOŚC = średni, duży)



Rys. 6.3. Przykład odpowiedzi

Otrzymany wynik pokazano graficznie na rys. 6.3. W ten sposób można obliczać odpowiedzi na pytania w języku przybliżonym systemu. Pytania te dotyczą informacji zawartej w systemie. Oczywiście większość pytań, które możemy zadawać w tak określonym języku przybliżonym, nie ma znaczenia praktycznego, metoda ta jednak pozwala na obliczanie odpowiedzi na niepełne dowolne pytania.

Praktycznie pytania w języku przybliżonym systemu mogą mieć znacznie prostszą postać. Załóżmy, że w kartotece kryminalnej mamy opisanych przestępców, z tym że nie wszystkie informacje w niej są dokładne. Możemy dokładnie nie znać wieku, koloru oczu, itd. znajdujących się tam przestępców. Możemy być zainteresowani wyszukaniem w kartotece wszystkich przestępców podejrzanych o dokonanie przestępstwa — o których wiadomo z zeznań świadków, że są wzrostu wysokiego, w średnim wieku oraz wiadomo, że mają oczy koloru niebieskiego lub szarego. Dla wyszukania więc wszystkich podejrzanych należy utworzyć pytanie

$$(\text{WZROST} = \text{duży}) \cdot (\text{WIEK} = \text{średni}) \cdot \bar{C}(\text{KOLOR OCZU} = \text{niebieski, szary})$$

W ten sposób z kartoteki wyszukamy wszystkie osoby wysokie w średnim wieku, u których nie wykluczamy koloru oczu niebieskiego bądź szarego. A więc jako podejrzanych musimy wziąć wszystkie osoby figurujące w kartotece wysokie w średnim wieku, o których wiemy, że mają kolor oczu niebieski, dalej osoby o których wiemy, że mają kolor oczu szary, następnie te osoby, o których wiemy, że mają kolor oczu niebieski lub szary, ale nie wiemy, który z nich, następnie osoby, o których wiemy, że mają kolor oczu niebieski lub piwny, ale nie wiemy, który z nich, dalej osoby, o których wiemy, że mają kolor oczu szary bądź piwny, ale również nie wiemy, który z nich itd. ∎

W ten sposób za pomocą języka przybliżonego można wyszukiwać obiekty, o których nie zdołano uzyskać dokładnej informacji, a jedynie informację przybliżoną.

6.7. OBLICZANIE PRZYBLIŻONE WARTOŚCI TERMÓW JĘZYKA DOKŁADNEGO

Przypominamy, że w języku dokładnym systemu nie występują ani deskryptory przybliżone, ani operacje $\underline{C}$, $\bar{C}$ (patrz p. 6.4). Jest to więc język taki sam, jak opisany w rozdziale drugim.

Jeżeli w termie języka dokładnego występuje atrybut przybliżony, to na podstawie podanych zasad nie potrafimy obliczyć wartości takiego termu w danym przybliżonym systemie informacyjnym. Na przykład, jeżeli w jakimś przybliżonym systemie informacyjnym atrybut KOLOR jest atrybutem przybliżonym, to nie można znaleźć wszystkich obiektów tego systemu, które mają np. kolor czerwony. Można natomiast wyszukać te obiekty systemu, o których wiadomo, że mają na pewno kolor czerwony oraz te, o których nie można wykluczyć, że mają kolor czerwony. W pierwszym przypadku otrzymany zbiór obiektów jest zawarty w zbiorze wszystkich czerwonych obiektów, ale mogą istnieć w systemie obiekty, które są czerwone, ale o tym nie wiemy. W drugim przypadku otrzymamy zbiór obiektów, który zawiera na pewno wszystkie obiekty, które w rzeczywistości mają kolor czerwony, ale znajdą się w nim również obiekty, które mogą się okazać innego koloru.

Inaczej mówiąc, nie można w systemie przybliżonym znaleźć wtedy odpowiedzi na pytanie

(KOLOR = czerwony)

natomiast można znaleźć odpowiedzi na pytania

$\underline{C}$(KOLOR = czerwony)

$\overline{C}$(KOLOR = czerwony)

Ogólnie, jeżeli t jest termem języka dokładnego systemu przybliżonego S, to $\underline{C}(t)$ oraz $\overline{C}(t)$ będziemy nazywać odpowiednio *przybliżeniem dolnym* i *górnym* termu t w języku przybliżonym systemu L_S, natomiast $\sigma_S(\underline{C}(t))$ oraz $\sigma_S(\overline{C}(t))$ będziemy nazywać *znaczeniem przybliżonym dolnym* i *górnym* termu t w systemie przybliżonym S.

Obliczanie obu przybliżeń odbywa się oczywiście na podstawie semantyki i aksjomatów języka przybliżonego.

Jeżeli term t spełnia warunki (7'), (11') podane w p. 6.6, to $\sigma_S(\underline{C}(t))$ oraz $\sigma_S(\overline{C}(t))$ są odpowiednio maksymalnym i minimalnym (w sensie zawierania) przybliżeniem dolnym i górnym $\sigma_S(t)$ w systemie S.

Jeżeli natomiast warunki te nie są spełnione, to posługując się regułami (7) i (11), otrzymamy również odpowiedzi przybliżone, gdyż reguła (7) pozwala na zastąpienie dolnego przybliżenia termu $t+t'$ zbiorem mniejszym, zaś reguła (11) pozwala na zastąpienie górnego przybliżenia termu $t \cdot t'$ zbiorem większym.

Przykład 6.2

Weźmy pod uwagę system przybliżony omawiany w p. 6.3, w którym atrybut KOLOR jest atrybutem przybliżonym, i obliczmy dolne i górne przybliżenie termu

(KOLOR = czerwony) + (KOLOR = niebieski)

Odpowiednie przybliżenia mają postać:

$\underline{C}$((KOLOR = czerwony) + (KOLOR = niebieski))

$\overline{C}$((KOLOR = czerwony) + (KOLOR = niebieski))

Na podstawie aksjomatów A1, oraz A4 (p. 6.5) otrzymamy

$\underline{C}$(KOLOR = czerwony, niebieski)

oraz

$\overline{C}$(KOLOR = czerwony, niebieski)

Następnie możemy obliczyć

$$\sigma(\overline{C}(\text{KOLOR} = \text{czerwony, niebieski})) =$$
$$E_{n,c} \cup E_n \cup E_c = \{x_2, x_4\} \cup \{x_1, x_3, x_{11}\} \cup \varnothing =$$
$$= \{x_1, x_2, x_3, x_4, x_{11}\}$$

oraz

$$\sigma(\bar{C}(\text{KOLOR} = \text{czerwony, niebieski})) =$$
$$E_{\mathrm{n,c}} \cup E_{\mathrm{n}} \cup E_{\mathrm{c}} \cup E_{\mathrm{n,z}} \cup E_{\mathrm{c,z}} \cup E_{\mathrm{n,c,z}} =$$
$$= \{x_1, x_2, x_3, x_4, x_5, x_7, x_8, x_9, x_{11}\} \quad \blacksquare$$

Pytanie nasze w języku dokładnym brzmiało: które z obiektów systemu są czerwone lub niebieskie. Na to pytanie w rozważanym systemie nie można dostać dokładnej odpowiedzi. Możemy natomiast stwierdzić, że obiekty $x_1, x_2, x_3, x_4, x_{11}$ są na pewno czerwone lub niebieskie (dolne przybliżenie) oraz, iż nie wykluczone, że obiekty x_5, x_7, x_8, x_9 są również czerwone albo niebieskie (razem z dolnym przybliżeniem dają one górne przybliżenie).

Nie będziemy tej sprawy omawiać dokładniej, gdyż jest ona oczywista. Nie będziemy tu omawiać bliżej również obliczania odpowiedzi przybliżonych, ponieważ sprowadza się ono do obliczania znaczenia termów języka przybliżonego, które było dokładnie omawiane w poprzednim punkcie.

7. STOCHASTYCZNE SYSTEMY INFORMACYJNE

7.1. WPROWADZENIE

W rozważanych do tej pory systemach informacyjnych występowała zawsze jedna z następujących sytuacji:

(a) wiedzieliśmy, że obiekt x ma daną własność,

(b) wiedzieliśmy, że obiekt x nie ma danej własności,

(c) nie wiedzieliśmy, czy obiekt x ma daną własność czy też nie.

W rozdziale tym będziemy omawiać systemy, w których dopuszczamy sytuację (c), z tym jednakże, że stopień naszej niewiedzy będzie wyrażalny pewną miarą, a mianowicie prawdopodobieństwem posiadania określonej własności. Tak więc będziemy mówić, że obiekt x ma własność (a, v) z prawdopodobieństwem $p (0 \leqslant \leqslant p \leqslant 1)$. Na przykład powiemy, że x jest czerwone z prawdopodobieństwem 1/2.

Bardziej interesujący jest przypadek, kiedy nie znamy konkretnego prawdopodobieństwa posiadania jakiejś własności, lecz znamy przedział, do którego to prawdopodobieństwo należy. Na przykład możemy wiedzieć, że prawdopodobieństwo, iż dany obiekt jest koloru czerwonego, jest większe od 1/2, tj. — należy do przedziału $(1/2, 1\rangle$. Rozważany na początku przypadek mieści się w ostatnim, gdyż prawdopodobieństwo p można traktować jako przedział $\langle p, p\rangle$.

Pytania w tego rodzaju systemach informacyjnych będą dotyczyły również podzbiorów obiektów, które mają określone własności z prawdopodobieństwem należącym do zadanego przedziału. Na przykład możemy być zainteresowani wyszukaniem wszystkich obiektów, które są koloru czerwonego z prawdopodobieństwem należącym do przedziału $\langle 1/4, 3/4\rangle$, tj. z prawdopodobieństwem nie mniejszym niż 1/4, ale nie większym niż 3/4.

Problemy tego typu występują w wielu dziedzinach, np. w medycynie, kryminalistyce, geologii, technice, archeologii itd., dlatego zajmowanie się tego rodzaju systemami jest uzasadnione praktycznie.

Systemy takie były rozpatrywane w pracach [66], [33], [82]. Podane tu podejście wywodzi się z identycznych założeń, jak we wspomnianych powyżej pracach, tutaj jednak kładziemy zasadniczy nacisk na rolę zbiorów elementarnych, stąd w naszym podejściu są uwypuklone inne aspekty niż w cytowanych pracach.

Jeszcze inne podejście do tego zagadnienia jest zaproponowane w pracy: Z. Pawlak — Partial Observability and Rough Probability, *Prace IPI PAN* No 470, 1983.

7.2. STOCHASTYCZNY SYSTEM INFORMACYJNY

Wprowadzone w tym punkcie pojęcie systemu informacyjnego będzie odpowiadało intuicji podanej we wstępie do tego rozdziału i będzie ono uogólnieniem rozważanych do tej pory systemów informacyjnych.

Definicja

Przez *stochastyczny system informacyjny* będziemy rozumieć czwórkę

$$S = \langle X, A, V, \pi \rangle$$

gdzie: X — zbiór obiektów, A — zbiór atrybutów, $V = \bigcup\limits_{a \in A} V_a$ — zbiór wartości atrybutów, $\pi\colon X \times A \times V \to \mathscr{P}(\langle 0, 1 \rangle)$ — funkcja informacji. ■

Zbiory X, A, V są tu identyczne jak w definicjach rozważanych poprzednio, natomiast zamiast funkcji ϱ występuje funkcja π, która każdemu obiektowi $x \in X$, atrybutowi $a \in A$, wartości $v \in V_a$ przypisuje podzbiór $P \subset \langle 0, 1 \rangle$. Równość $\pi(x, a, v) = P$ oznacza, że prawdopodobieństwo tego, że x ma własność $v \in V_a$, należy do zbioru P, tj. $p(x, a, v) \in \pi(x, a, v)$, gdzie $p(x, a, v)$ jest prawdopodobieństwem posiadania przez x własności (a, v).

Definicja

Funkcję

$$\pi_x\colon A \times V \to \mathscr{P}(\langle 0, 1 \rangle)$$

taką, że

$$\pi_x(a, v) = \pi(x, a, v) \text{ dla każdego } x \in X,\ a \in A,\ v \in V_a$$

będziemy nazywali *informacją o obiekcie* x w systemie S. ■

Ponieważ prawdopodobieństwo $p(x, a, v)$ musi spełniać warunek

(a) $\sum\limits_{v \in V_a} p(x, a, v) = 1$ dla każdego $x \in X$,

funkcja π musi spełniać dwa następujące warunki:

(b) $\sum\limits_{v \in V_a} \pi^*(x, a, v) \geqslant 1$ dla każdego $x \in X$

(c) $\sum\limits_{v \in V_a} \pi_*(x, a, v) \leqslant 1$ dla każdego $x \in X$

gdzie $\pi^*(x, a, v)$ oraz $\pi_*(x, a, v)$ oznaczają odpowiednio górny i dolny kres zbioru $\pi(x, a, v)$.

Przykład 7.1

Rozważmy bardzo prosty stochastyczny system informacyjny posiadający osiem obiektów i dwa atrybuty a, b, przy czym $V_a = \{v_1, v_2, v_3\}$ oraz $V_b = \{u_1, u_2\}$.

System ten jest określony tabelką:

	(a, v_1)	(a, v_2)	(a, v_3)	(b, u_1)	(b, u_2)
x_1	$\langle 0, 1/3\rangle$	$\langle 1/2, 1\rangle$	$\langle 1/4, 1\rangle$	$\langle 1/6, 1/4\rangle$	$\langle 1/4, 1\rangle$
x_2	$\langle 1/2, 1\rangle$	$\langle 0, 3/4\rangle$	$\langle 1/4, 1\rangle$	$\langle 1/4, 1\rangle$	$\langle 0, 1/3\rangle$
x_3	$\langle 0, 1/3\rangle$	$\langle 1/2, 1\rangle$	$\langle 1/4, 1\rangle$	$\langle 1/6, 1/4\rangle$	$\langle 1/4, 1\rangle$
x_4	$\langle 0, 1/3\rangle$	$\langle 1/4, 1\rangle$	$\langle 0, 1/2\rangle$	$\langle 1/3, 3/4\rangle$	$\langle 0, 1/4\rangle$
x_5	$\langle 1/2, 1\rangle$	$\langle 0, 3/4\rangle$	$\langle 1/4, 1\rangle$	$\langle 1/4, 1\rangle$	$\langle 0, 1/3\rangle$
x_6	$\langle 0, 1/3\rangle$	$\langle 1/2, 1\rangle$	$\langle 1/4, 1\rangle$	$\langle 1/6, 1/4\rangle$	$\langle 1/4, 1\rangle$
x_7	$\langle 0, 1/3\rangle$	$\langle 1/4, 1\rangle$	$\langle 0, 1/2\rangle$	$\langle 1/3, 3/4\rangle$	$\langle 0, 1/4\rangle$
x_8	$\langle 1/2, 1\rangle$	$\langle 0, 3/4\rangle$	$\langle 1/4, 1\rangle$	$\langle 1/4, 1\rangle$	$\langle 0, 1/3\rangle$

■

Definicja

Obiekty $x, y \in X$ są w systemie $S = \langle X, A, V, \pi\rangle$ *nierozróżnialne ze względu na własność* (a, v) wtedy i tylko wtedy, gdy

$$\pi_x(a, v) = \pi_y(a, v) \qquad ■$$

Definicja

Obiekty $x, y \in X$ są nierozróżnialne w systemie $S = \langle X, A, V, \pi\rangle$ *ze względu na zbiór własności* $W = \{(a, v)\colon a \in B, B \subset A, v \in V_a\}$ wtedy i tylko wtedy, gdy

$$\pi_x(a, v) = \pi_y(a, v), \text{ dla każdego } (a, v) \in W \qquad ■$$

Definicja

Obiekty $x, y \in X$ są *nierozróżnialne w systemie* $S = \langle X, A, V, \pi\rangle$ wtedy i tylko wtedy, gdy $\pi_x(a, v) = \pi_y(a, v)$ dla każdej pary (a, v), $a \in A$, $v \in V_a$, tj. gdy

$$\pi_x = \pi_y \qquad ■$$

Łatwo sprawdzić, że wszystkie trzy relacje nierozróżnialności są relacjami równoważności, np.

(1) $\pi_x = \pi_x$,

(2) Jeżeli $\pi_x = \pi_y$ to $\pi_y = \pi_x$,

(3) Jeżeli $\pi_x = \pi_y$ oraz $\pi_y = \pi_z$, to $\pi_x = \pi_z$.

Jeżeli x jest równoważne y ze względu na własność (a, v), zapiszemy $x \underset{(a,v)}{\sim} y$, natomiast jeżeli x, y są równoważne ze względu na każdą własność w systemie S, zapiszemy $x \underset{S}{\sim} y$, lub krótko $x \sim y$, gdy system jest znany.

Podobnie jak w systemach informacyjnych rozważanych poprzednio

$$\underset{S}{\sim} = \bigcap_{(a,v) \in A \times V} \underset{(a,v)}{\sim}$$

Zbiorami elementarnymi w systemie stochastycznym są więc również, jak w poprzednich przypadkach, klasy abstrakcji relacji $\underset{S}{\sim}$.

Na przykład w podanym przykładzie stochastycznym systemu informacyjnego zbiorami elementarnymi są:

$$\{x_1, x_3, x_6\},\ \{x_2, x_5, x_8\},\ \{x_4, x_7\}$$

a relacja równoważności $\underset{(a,v_1)}{\sim}$ ma następujące klasy abstrakcji:

$$\{x_1,\ x_3,\ x_4,\ x_6,\ x_7\},\ \{x_2,\ x_8\}$$

natomiast relacja $\underset{(a,v_3)}{\sim}$ daje podział:

$$\{x_1, x_2, x_3, x_5, x_6, x_8\},\ \{x_4, x_7\}$$

Tak więc stochastyczne systemy informacyjne nie różnią się istotnie (są one szczególnym przypadkiem) z formalnego punktu widzenia od systemów informacyjnych rozważanych do tej pory. Możemy więc wszystkie wprowadzone poprzednio własności systemów informacyjnych przenieść po odpowiedniej interpretacji na systemy stochastyczne. Jeżeli pary (a, v) w systemach stochastycznych będziemy traktować jak atrybuty, prawdopodobieństwa zaś jak wartości tych atrybutów, to stochastyczny system informacyjny możemy traktować tak samo, jak przybliżony system informacyjny. Zbiór $\pi(x, a, v)$, do którego należy prawdopodobieństwo $p(x, a, v)$, możemy traktować identycznie jak informacje o obiekcie w systemach informacyjnych. Dlatego nie będziemy szczegółowo rozważali tutaj systemów stochastycznych.

7.3. JĘZYK STOCHASTYCZNYCH SYSTEMÓW INFORMACYJNYCH

Pytania w stochastycznym systemie informacyjnym będą dotyczyły obiektów mających określoną własność z prawdopodobieństwem należącym do zadanego podzbioru. Język systemu stochastycznego zbudujemy podobnie, jak to czyniliśmy poprzednio. Wyrażenia tego języka (termy) będą oznaczały podzbiory zbioru obiektów. Ponieważ stochastyczny system informacyjny można traktować jako szczególny przypadek przybliżonego systemu informacyjnego; tutaj, podobnie jak w systemach przybliżonych, zostaną wprowadzone operatory $\underline{C}$, $\overline{C}$ mówiące, że interesujące nas prawdopodobieństwo na pewno należy do zadanego zbioru P, bądź też jest nie wykluczone, że należy ono do zbioru P. Dlatego język systemów stochastycznych będzie przypominał język systemów przybliżonych.

Jako symbole języka stochastycznego przyjmiemy:

(1) stałe 0, 1;

(2) deskryptory stochastyczne (a, v, P), $a \in A$, $v \in V_a$, $P \subset \langle 0, 1 \rangle$;

(3) symbole operacji boolowskich $+$, $\cdot$, $\sim$;

(4) symbole $\underline{C}$, $\overline{C}$.

Dla uproszczenia pomijamy tu symbole $\rightarrow$, $\leftrightarrow$ używane w dotychczas rozważanych językach.

Wyrażenia postaci (a, v, P), $a \in A$, $v \in V_a$, $P \subset \langle 0, 1\rangle$ będziemy nazywać *deskryptorami stochastycznymi.*

Termy języka stochastycznego określimy indukcyjnie:

(1) Termami są stałe 0, 1

(2) Termami są wyrażenia (a, v, P), $\underline{C}(a, v, P)$, $\overline{C}(a, v, P)$, $a \in A$, $v \in V_a$, $P \subset \langle 0, 1\rangle$.

(3) Jeżeli t, t' są termami, to $(t \cdot t')$, $(t+t')$, $\sim t$ są również termami.

Na przykład term

(KOLOR, czerwony, $(1/2, 1\rangle$)

oznacza pytanie dotyczące obiektów, dla których prawdopodobieństwo, że kolor jest czerwony jest większe niż 1/2.

Semantyka tego języka jest zdefiniowana indukcyjnie w następujący sposób:

(1) $\sigma(0) = \emptyset$, $\sigma(1) = X$;

(2) $\sigma(a, v, P) = \{x \in X\colon \pi(x, a, v) = P\}$;

(3) $\sigma(\overline{C}(a, v, P)) = \{x \in X\colon \pi(x, a, v) \cap P \neq \emptyset\}$;

(4) $\sigma(\underline{C}(a, v, P)) = \{x \in X\colon \pi(x, a, v) \subset P\}$;

(5) $\sigma(t+t') = \sigma(t) \cup \sigma(t')$;

(6) $\sigma(t \cdot t') = \sigma(t) \cap \sigma(t')$;

(7) $\sigma(\sim t) = X - \sigma(t)$.

Dla uproszczenia przyjeliśmy tutaj, że operatory $\underline{C}$, $\overline{C}$ mogą występować tylko bezpośrednio przed deskryptorami stochastycznymi. Można oczywiście zdefiniować taki język, w którym operatory $\underline{C}$, $\overline{C}$ mogą być umieszczone dowolnie, podobnie jak w językach systemów przybliżonych, jednakże język taki nie ma wyraźnej interpretacji praktycznej, bowiem nie wiadomo, jaki byłby sens praktyczny tego rodzaju pytań.

Semantyka ta jest oczywista. Pewnego komentarza wymagają jedynie reguły (2), (3) i (4).

Znaczeniem deskryptora stochastycznego (a, v, P) jest klasa abstrakcji relacji $\underset{(a,v)}{\sim}$, tzn. zbiór tych wszystkich obiektów systemu, dla których prawdopodobieństwo posiadania własności (a, v) należy do zbioru P.

Znaczeniem wyrażenia $\underline{C}(a, v, P)$ jest suma tych wszystkich klas abstrakcji $\sigma(a, v, Q)$ relacji $\underset{(a,v)}{\sim}$, dla których $P \cap Q \neq \emptyset$.

I wreszcie znaczeniem wyrażenia $\underline{C}(a, v, P)$ jest suma tych wszystkich klas abstrakcji $\sigma(a, v, Q)$ relacji $\underset{(a,v)}{\sim}$, dla których $Q \subset P$.

Można więc napisać:

(a) $\bar{C}(a, v, P) = \sum_{Q \cap P \neq \emptyset} (a, v, Q)$

(b) $\underline{C}(a, v, P) = \sum_{Q \subset P} (a, v, Q)$

Wyrażenie $\bar{C}(a, v, P)$ należy więc czytać: prawdopodobieństwo własności (a, v) należy być może do zbioru P (tzn. nie jest wykluczone, że należy ono do P), wyrażenie $\underline{C}(a, v, P)$ należy czytać: prawdopodobieństwo własności (a, v) należy na pewno do zbioru P.

Zwróćmy uwagę, że w świetle podanych definicji, aby otrzymać wszystkie obiekty, które mają własność (a, v) z prawdopodobieństwem należącym do zbioru P, musimy wziąć nie tylko obiekty należące do zbioru $\sigma(a, v, P)$, ale również musimy dodać do odpowiedzi wszystkie zbiory $\sigma(a, v, Q)$, takie że $Q \subset P$. To znaczy, że jeżeli mamy np. w naszym systemie informacyjnym zbiory określone deskryptorami stochastycznymi

(KOLOR, czerwony, $\langle 1/2, 1 \rangle$) oraz (KOLOR, czerwony, $\langle 3/4, 1 \rangle$)

to dla otrzymania odpowiedzi na pytanie

$\underline{C}$(KOLOR, czerwony, $\langle 1/2, 1 \rangle$)

musimy wziąć sumę zbiorów

σ(KOLOR, czerwony, $\langle 1/2, 1 \rangle$)$\cup\sigma$(KOLOR, czerwony, $\langle 3/4, 1 \rangle$)

Z definicji operatorów $\underline{C}$, $\bar{C}$ wynika, że są one skrótami, które można z języka wyeliminować.

Dla uproszczenia wyrażeń języka stochastycznego będziemy używać aksjomatów algebry Boole'a oraz następujących aksjomatów charakteryzujących deskryptory stochastyczne:

A1. $(a, v, P) \cdot (a, v, Q) = 0$, gdy $P \neq Q$;

A2. $\sum_{P \subset \langle 0,1 \rangle} (a, v, P) = 1$;

A3. $\sim (a, v, P) = (a, v, \langle 0, 1 \rangle - P)$;

A4. $\underline{C}(a, v, P) + \underline{C}(a, v, Q) \subset \underline{C}(a, v, P \cup Q)$;

A5. $\underline{C}(a, v, P) \cdot \underline{C}(a, v, Q) = \underline{C}(a, v, P \cap Q)$;

A6. $\bar{C}(a, v, P) + \bar{C}(a, v, Q) = \bar{C}(a, v, P \cup Q)$;

A7. $\bar{C}(a, v, P) \cdot \bar{C}(a, v, Q) \supset \bar{C}(a, v, P \cap Q)$;

A8. $\sim \underline{C}(a, v, P) = \bar{C}(a, v, \langle 0, 1 \rangle - P)$;

A9. $\sim \bar{C}(a, v, P) = \underline{C}(a, v, \langle 0, 1 \rangle - P)$.

Aksjomaty A1÷A3 są odpowiednikami aksjomatów w systemie jednowartościowym. Aksjomat A1 mówi, że iloczyn dwu różnych deskryptorów stochastycznych o tych samych atrybutach i wartościach jest pusty. Wynika to z faktu, że deskryptory takie opisują różne klasy abstrakcji tej samej relacji.

Aksjomat A2 mówi, że suma wszystkich klas abstrakcji relacji $\underset{(a,v)}{\sim}$ jest równa całemu zbiorowi X, tzn. że o każdym obiekcie w systemie mamy informacje odnośnie do własności (a, v). Suma w tym aksjomacie jest brana nie po wszystkich podzbiorach przedziału $\langle 0, 1\rangle$, lecz tylko po tych podprzedziałach, które występują w danym systemie informacyjnym, a więc jest to suma skończona.

Następny aksjomat charakteryzuje negację i jest on oczywisty.

Pozostałe aksjomaty charakteryzują operatory $\underline{C}$ i $\bar{C}$. Są one oczywiste w świetle rozważań na temat systemów przybliżonych.

Dla przykładu obliczymy wartości kilku prostych termów, posługując się stochastycznym systemem informacyjnym podanym w przykładzie 7.1.

Bezpośrednio z definicji semantyki można obliczyć wartości następujących termów:

(1) $\sigma(a, v_2, \langle 1/4, 1\rangle) = \{x_4, x_7\}$

(2) $\sigma(\underline{C}(a, v_2, \langle 1/4, 1\rangle)) = \sigma(a, v_2, \langle 1/4, 1\rangle) \cup \sigma(a, v_2, \langle 1/2, 1\rangle) =$

$= \{x_4, x_7\} \cup \{x_1, x_3, x_6\} = \{x_1, x_3, x_4, x_6, x_7\}$

(3) $\sigma(\bar{C}(a, v_2, \langle 1/4, 1\rangle)) = \sigma(a, v_2, \langle 1/4, 1\rangle) \cup \sigma(a, v_2, \langle 1/2, 1\rangle) \cup$

$\cup\, \sigma(a, v_2, \langle 0, 3/4\rangle) = \{x_4, x_7\} \cup \{x_1, x_3, x_6\} \cup \{x_2, x_5, x_8\} = X$

(4) $\sigma(a, v_1, \langle 0, 1/10\rangle) = \emptyset$

(5) $\sigma(\underline{C}(a, v_1, \langle 0, 1/10\rangle)) = \emptyset$

(6) $\sigma(\bar{C}(a, v_1, \langle 0, 1/10\rangle)) = \sigma(a, v_1, \langle 0, 1/3\rangle) = \{x_1, x_3, x_6\}$

Rozpatrzymy teraz nieco bardziej złożone termy:

(7) $\sigma(\sim \underline{C}(a, v_1, (1/5, 4/(5))) = \sigma(\bar{C}(a, v_1, \langle 0, 1/5\rangle \cup \langle 4/5, 1\rangle))$

na podstawie aksjomatu A8, skąd na podstawie aksjomatu A6 otrzymamy

$\sigma(\bar{C}(a, v_1, \langle 0, 1/5\rangle)) \cup \sigma(\bar{C}(a, v_1, \langle 4/5, 1\rangle))$

skąd na podstawie definicji operatora $\bar{C}$ otrzymamy

$\sigma(\bar{C}(a, v_1, \langle 0, 1/5\rangle)) = \sigma(a, v_1, \langle 0, 1/3\rangle) = \{x_1, x_3, x_6\}$

zaś

$\sigma(\bar{C}(a, v_1, \langle 4/5, 1\rangle)) = \sigma(a, v_1, \langle 1/2, 1\rangle) = \{x_2, x_5, x_8\}$

skąd ostatecznie otrzymamy rezultat końcowy

$\{x_1, x_2, x_3, x_5, x_6, x_8\}$

Obliczmy

$$(8)\ \sigma(\sim \bar{C}(a, v_1, (1/4, 3/14)))$$

Stosując aksjomat A9, otrzymamy

$$\sigma(\underline{C}(a, v_1, \langle 0, 1/4\rangle \cup \langle 3/4, 1\rangle))$$

Łatwo zauważyć, że jeżeli P i Q są rozłączne, to w aksjomacie A4 znak zawierania można zastąpić znakiem równości. Stosując aksjomat A4 do ostatniego termu, otrzymamy

$$\sigma(\underline{C}(a, v_1, \langle 0, 1/4\rangle)) \cup \sigma(\underline{C}(a, v_1, \langle 3/4, 1\rangle))$$

Ponieważ

$$\sigma(\underline{C}(a, v_1, \langle 0, 1/4\rangle)) = \emptyset$$

oraz

$$\sigma(\underline{C}(a, v_1, \langle 3/4, 1\rangle)) = \emptyset$$

więc wartość obliczanego termu jest zbiorem pustym.

Obliczanie termów zawierających symbole operacji $+, \cdot$ jest proste i nie sprawia trudności, dlatego nie będziemy ich tutaj bliżej rozpatrywać.

7.4. PODSUMOWANIE

We wszystkich rozpatrywanych poprzednio systemach informacyjnych zasadniczą rolę grały zbiory elementarne. Zbiory te były klasami abstrakcji relacji równoważności generowanej w każdym systemie informacyjnym przez pojęcie obiektów nierozróżnialnych w systemie. Tak więc zamiast przyjmować różne definicje dla różnych klas systemów informacyjnych, można zdefiniować każdy z rozważanych do tej pory systemów informacyjnych jako parę

$$S = \langle X, R\rangle$$

gdzie: x — zbiór obiektów systemu, R — relacja równoważności określona na X. Jeżeli $x, y \in X$ oraz x i y są w relacji R, to obiekty x, y są nierozróżnialne w systemie S.

Język L_S systemu S będzie określany podobnie jak poprzednio. Alfabet języka L_S składał się z symboli:

(1) stałych $0, 1$;

(2) stałych $e_1, e_2, \ldots, e_k$;

(3) symboli operacji $+, \cdot, \sim$;

(4) nawiasów (,).

Termy języka L_S są określone indukcyjnie, jak następuje:

(1) Stałe $0, 1$ oraz $e_1, e_2, \ldots, e_k$ są termami.

(2) Jeżeli t, t' są termami, to również wyrażenia $(t+t')$, $(t \cdot t')$, $\sim t$ są termami.

Semantyka języka L_S jest zdefiniowana następująco:

(1) $\sigma_S(0) = \emptyset, \quad \sigma_S(1) = X$

(2) $\sigma_S(e_i) = E_i, \quad E_i$ — zbiór elementarny

(3) $\sigma_S(t+t') = \sigma_S(t) \cup \sigma_S(t')$

(4) $\sigma_S(t \cdot t') = \sigma_S(t) \cap \sigma_S(t')$

(5) $\sigma_S(\sim t) = X - \sigma_S(t)$

Stałe e_i traktujemy jako nazwy zbiorów elementarnych systemu.

W celu określenia różnych klas systemów informacyjnych rozpatrywanych poprzednio, należy jedynie rozszerzyć język L_S w ten sposób, aby można było zdefiniować stałe $e_1, e_2, \ldots, e_k$ będące nazwami zbiorów elementarnych, specyficznych dla danego rodzaju systemu informacyjnego, tzn. stałe $e_1, e_2, \ldots, e_k$ będą dla jednowartościowych systemów informacyjnych wyrażeniami postaci:

$$(a_1, v_1) \cdot (a_2, v_2) \cdot \ldots \cdot (a_n, v_n)$$

gdzie $v_i \in V_{a_i}$.

Dla systemów informacyjnych wielowartościowych oraz przybliżonych stałe będą miały postać

$$(a_1, U_1) \cdot (a_2, U_2) \cdot \ldots \cdot (a_n, U_n)$$

gdzie $U_i \subset V_{a_i}$.

Dla systemów stochastycznych stałe są wyrażeniami postaci:

$$(a_1, v_1, P_1) \cdot (a_2, v_2, P_2) \cdot \ldots \cdot (a_n, v_n, P_n)$$

gdzie $v_i \in V_{a_i}$ oraz $P_i \subset \langle 0, 1 \rangle$.

W rezultacie wiele podstawowych własności systemów informacyjnych i ich języków można przeanalizować na modelu ogólnym systemu (np. postać normalna, równoważność systemów itd.), natomiast własności poszczególnych rodzajów systemów można badać, opierając się na aksjomatach specyficznych systemów, charakteryzujących w istocie strukturę ich zbiorów elementarnych.

Podejście takie jest uzasadnione nie tylko względami teoretycznymi, lecz także praktycznymi. Zbiory elementarne stanowią bowiem podstawową strukturę charakteryzującą organizację danych w pamięci maszyny liczącej, w istocie jednakowe dla różnych rodzajów systemów informacyjnych. Natomiast sposób ich tworzenia i wykorzystania jest związany ze specyfiką systemu informacyjnego, która się wyraża w języku systemu. Język ten z jednej strony służy do opisu danych, poprzez charakterystykę zbiorów elementarnych — z drugiej zaś określa mechanizm znajdowania odpowiedzi na pytania, tj. mechanizm operowania zbiorami elementarnymi celem uzyskania odpowiedzi na zadawane pytania.

8. PRZYBLIŻONA KLASYFIKACJA OBIEKTÓW

8.1. WPROWADZENIE

W poprzednich rozdziałach zajmowaliśmy się następującym problemem: dane są system informacyjny $S = \langle X, A, V, \varrho \rangle$, język L_S oraz term t należący do języka L_S, znaleźć zbiór obiektów systemu S opisanych przez term t.

Teraz zajmiemy się problemem który można sformułować następująco: dane są system informacyjny $S = \langle X, A, V, \varrho \rangle$, język L_S oraz pewien zbiór $Y \subset X$ obiektów systemu S, znaleźć term t języka L_S, taki że t jest opisem przybliżonym zbioru Y.

Pierwszy problem jest podstawą systemów wyszukiwania informacji, drugi zaś — tzw. sztucznej inteligencji, co wyjaśnimy bliżej na przykładzie.

Wyobraźmy sobie, że mamy zbiór zdjęć rentgenowskich klatki piersiowej pacjentów podejrzanych o chorobę płuc. Specjalista lekarz na podstawie swej wiedzy, obserwując kolejno wszystkie zdjęcia, klasyfikuje je na te, które wskazują istnienie choroby, i te, na których oznak choroby nie ma. Ściślej, na podstawie tych zdjęć klasyfikuje się pacjentów na chorych i zdrowych. Z taką sytuacją mamy do czynienia np. przy masowych prześwietleniach ludności przeprowadzanych co jakiś czas celem wczesnego wykrycia chorób klatki piersiowej.

Lekarz klasyfikujący zdjęcia nie musi bliżej motywować swych decyzji, klasyfikacji dokonuje na podstawie swej wiedzy medycznej i doświadczenia. Gdyby jednak chciał on nauczyć kogoś (np. studentów medycyny) rozpoznawać osoby chore na podstawie zdjęć rentgenowskich, musiałby umieć scharakteryzować swoje decyzje, podając że osoby chore to takie, których zdjęcia rentgenowskie mają określone własności, np. występują na nich deformacje niektórych organów, plamy, cienie itd. Gdyby opis taki udało się wykonać, wtedy student medycyny (czy też dowolna osoba) nie mająca wiedzy eksperta ani jego doświadczenia mógłby trafnie klasyfikować zdjęcia rentgenowskie jedynie na podstawie stwierdzenia, czy mają one odpowiednie własności podane przez eskperta czy też nie.

Podobny charakter ma nie tylko klasyfikacja zdjęć rentgenowskich, ale dowolna diagnoza medyczna.

Takie same problemy występują nie tylko w medycynie, lecz także w wielu innych dziedzinach. Ich istotą jest opisanie zadanego zbioru obiektów w odpowiednim języku formalnym.

Zadanie to jest ściśle związane z tzw. logiką indukcji. Głównym celem badań tej logiki jest analiza stawiania hipotez na podstawie przykładów. Rozpatrywana na początku tego punktu diagnoza medyczna jest właśnie typowym przykładem rozumowania indukcyjnego, w podanych bowiem przykładach osób chorych (przypadki rozpoznane przez eksperta) pytamy — jakie są charakterystyczne symptomy rozpoznawanej choroby.

Istnieje bogata literatura dotycząca logiki indukcji (patrz np. prace [13], [14], [15]) oraz jej zastosowań (np. prace [12], [78], [79], [80]).

W niniejszej książce zajmiemy się analizą tego zagadnienia w oparciu o pojęcie systemu informacyjnego oraz jego języka, podane w pierwszym i drugim rozdziale oraz pojęcie zbioru przybliżonego wprowadzone w pracy [86], oraz badane w pracach [87], [34], [35], [85].

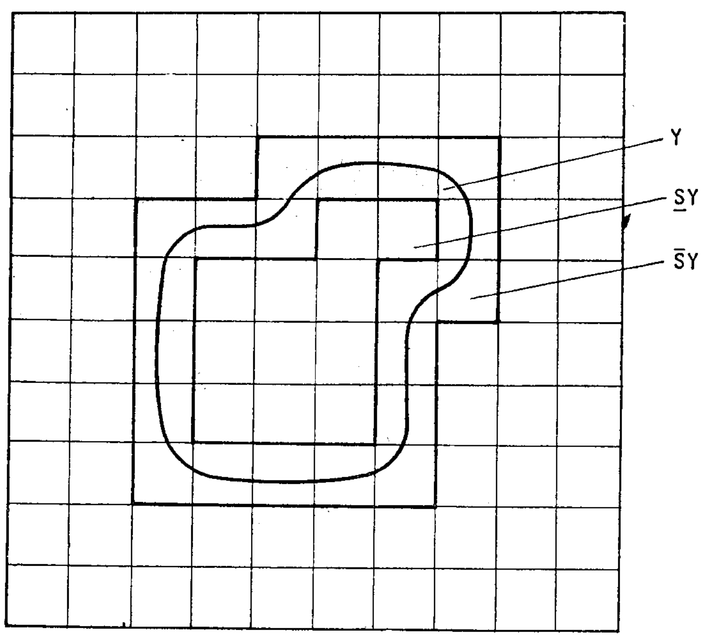
8.2. ZBIORY PRZYBLIŻONE

Podstawową rolę w naszym podejściu do rozpatrywanego zagadnienia będzie odgrywać pojęcie zbioru przybliżonego zdefiniowane i omówione w niniejszym punkcie[1].

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym oraz niech $Y \subset X$.

Największy (w sensie zawierania) zbiór opisywalny w systemie S zawarty w Y nazwiemy *dolnym przybliżeniem zbioru Y w systemie S* i oznaczymy przez $\underline{S}Y$.

Najmniejszy (w sensie zawierania) zbiór opisywalny w systemie S zawierający zbiór Y nazwiemy *górnym przybliżeniem zbioru Y w systemie S* i oznaczymy przez $\bar{S}Y$.

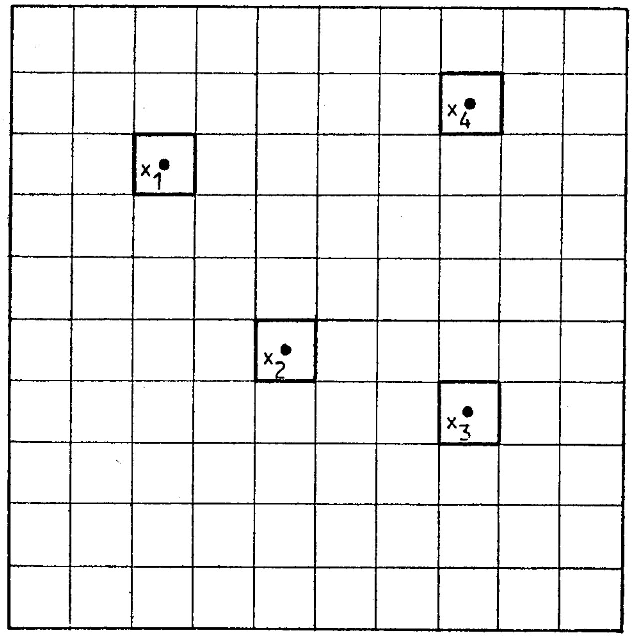


Rys. 8.1. Dolne i górne przybliżenie zbioru

Pojęcia te były już wprowadzone w p. 6.3, jednakże obecnie omówimy je dokładniej.

Na rysunku 8.1 przedstawiono ilustrację przybliżenia dolnego i górnego pewnego zbioru w systemie informacyjnym, którego zbiory elementarne są kwadratami wyznaczonymi przez siatkę. Rysunek jest oczywisty i nie wymaga bliższego objaśnienia.

[1] Ogólniejsze sformułowanie pojęcia zbioru przybliżonego można znaleźć w pracy: Z. Pawlak — Rough sets, *International Journal of Computer and Information Sciences*, 1982, 11 (5), s. 341—356.



Rys. 8.2. Przykład przybliżeń

Inny przykład przybliżeń pokazano na rys. 8.2. Przybliżeniem górnym zbioru $Y=\{x_1, x_2, x_3, x_4\}$ jest zbiór będący sumą zbiorów elementarnych zawierających punkty x_1, x_2, x_3, x_4, zaznaczone na rysunku grubymi liniami. Dolnym przybliżeniem tego zbioru jest zbiór pusty.

Przykład 8.1

Niech system informacyjny $S=\langle X, A, V, \varrho\rangle$ będzie określony tabelą

X	a	b	c
x_1	u_1	v_1	w_1
x_2	u_2	v_1	w_2
x_3	u_1	v_2	w_2
x_4	u_1	v_1	w_1
x_5	u_2	v_1	w_2
x_6	u_2	v_2	w_1
x_7	u_2	v_1	w_2
x_8	u_1	v_1	w_1

System ten ma następujące zbiory elementarne:

$$E_1=\{x_1, x_4, x_8\}$$

$$E_2=\{x_2, x_5, x_7\}$$

$$E_3=\{x_3\}$$

$$E_4=\{x_6\}$$

Następujące zbiory:

$$Y_1=\{x_1, x_4, x_5\}$$

$$Y_2=\{x_3, x_5\}$$

$$Y_3=\{x_3, x_6, x_8\}$$

$$Y_4=\{x_1, x_4, x_5, x_6\}$$

$$Y_5=\{x_1, x_2, x_3, x_5, x_7\}$$

będą miały przybliżenia podane poniżej:

$$\bar{S}Y_1=E_1\cup E_2$$

$$\bar{S}Y_2=E_2\cup E_3$$

$$\bar{S}Y_3=E_1\cup E_3\cup E_4$$

$$\bar{S}Y_4=E_1\cup E_2\cup E_4$$

$$\bar{S}Y_5=E_1\cup E_2\cup E_3$$

$$\underline{S}Y_1=\emptyset$$

$$\underline{S}Y_2=E_3$$

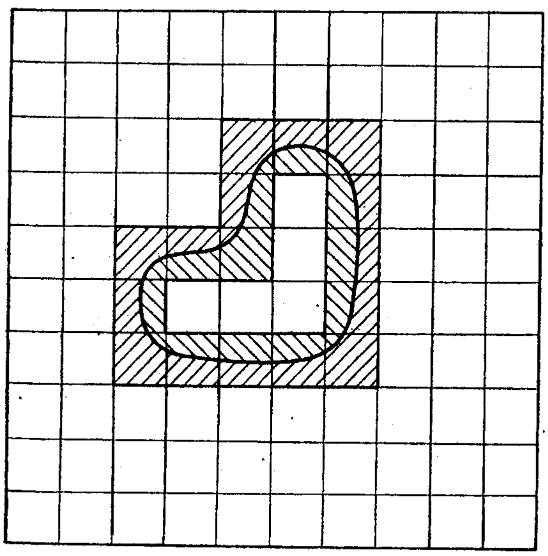
$$\underline{S}Y_3=E_3\cup E_4$$

$$\underline{S}Y_4=E_4$$

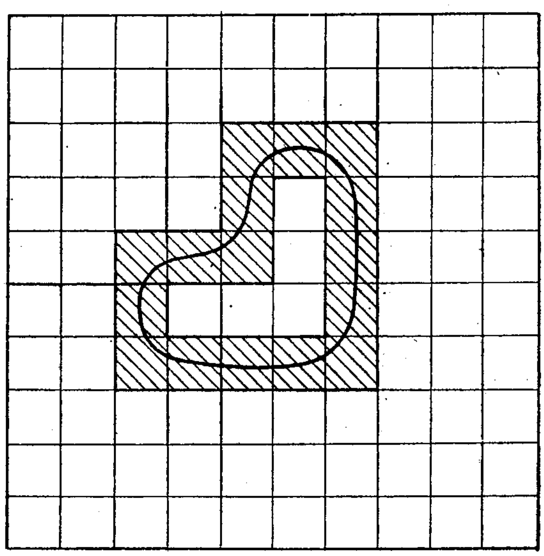
$$\underline{S}Y_5=E_2\cup E_3 \qquad \blacksquare$$

Wprowadzimy jeszcze kilka potrzebnych nam pojęć.

Brzegiem wewnętrznym zbioru $Y\subset X$ w systemie informacyjnym S będziemy nazywali zbiór $\underline{E}_S(Y)=Y-\underline{S}Y$, *brzegiem zewnętrznym zbioru* $Y\subset X$ w systemie S nazwiemy zbiór $\bar{E}_S(Y)=\bar{S}Y-Y$.



Rys. 8.3. Brzeg zbioru



Rys. 8.4. Ograniczenie zbioru

Ograniczeniem zbioru $Y \subset X$ w systemie S nazwiemy zbiór $B_S(Y) = \bar{S}Y - \underline{S}Y$. Oczywiście $B_S(Y) = \underline{E}_S(Y) \cup \bar{E}_S(Y)$. Pojęcia te mają naturalną interpretację geometryczną, pokazaną na rysunkach 8.3 i 8.4.

Dla zbiorów Y_1, Y_2, Y_3, Y_4, Y_5 rozpatrywanych w przykładzie 8.1 ograniczenia i brzegi są następujące:

$$B_S(Y_1) = E_1 \cup E_2$$

$$B_S(Y_2) = (E_2 \cup E_3) - E_3 = E_2$$

$$B_S(Y_3) = (E_1 \cup E_3 \cup E_4) - (E_3 \cup E_4) = E_1$$

$$B_S(Y_4) = (E_1 \cup E_2 \cup E_4) - E_4 = E_1 \cup E_2$$

$$B_S(Y_5) = (E_1 \cup E_2 \cup E_3) - (E_2 \cup E_3) = E_1$$

$$\underline{E}_S(Y_1) = Y_1 - \emptyset = Y_1$$

$$\underline{E}_S(Y_2) = Y_2 - E_3 = \{x_3, x_5\} - \{x_3\} = x_5$$

$$\underline{E}_S(Y_3) = Y_3 - \{E_3 \cup E_4\} = \{x_3, x_6, x_8\} - \{x_3, x_6\} = \{x_8\}$$

$$\underline{E}_S(Y_4) = Y_4 - E_4 = \{x_1, x_4, x_5, x_6\} - \{x_6\} = \{x_1, x_4, x_5\}$$

$$\underline{E}_S(Y_5) = Y_5 - \{E_2 \cup E_3\} = \{x_1, x_2, x_3, x_5, x_7\} - \{x_2, x_3, x_5, x_7\} = \{x_1\}$$

$$\bar{E}_S(Y_1) = (E_1 \cup E_2) - \{x_1, x_4, x_5\} = \{x_2, x_7, x_8\}$$

$$\bar{E}_S(Y_2) = (E_2 \cup E_3) - \{x_3, x_5\} = \{x_2, x_7\}$$

$$\bar{E}_S(Y_3) = (E_1 \cup E_3 \cup E_4) - \{x_6, x_3, x_8\} = \{x_1, x_4\}$$

$$\bar{E}_S(Y_4) = (E_1 \cup E_2 \cup E_4) - \{x_1, x_4, x_5, x_6\} = \{x_2, x_7, x_8\}$$

$$\bar{E}_S(Y_5) = (E_1 \cup E_2 \cup E_3) - \{x_1, x_2, x_3, x_5, x_7\} = \{x_4, x_8\}$$

■

Wprowadzimy ponadto pojęcie dokładności przybliżenia:

$$\bar{\eta}_S(Y) = \frac{\mathrm{card}(Y)}{\mathrm{card}(\bar{S}Y)}$$

$$\underline{\eta}_S(Y) = \frac{\mathrm{card}(\underline{S}Y)}{\mathrm{card}(Y)}$$

Dokładność górnego przybliżenia zbioru Y w systemie S $(\bar{\eta}_S(Y))$ oraz dokładność dolnego przybliżenia zbioru Y w systemie S $(\underline{\eta}_S(y))$ są więc liczbami należącymi do przedziału $\langle 0, 1 \rangle$. Sens obu tych pojęć jest intuicyjnie oczywisty.

Rozpatrywane w poprzednim przykładzie zbiory będą miały następujące dokładności przybliżeń górnych i dolnych:

$$\bar{\eta}_S(Y_1) = \frac{3}{6} = \frac{1}{2} \qquad \underline{\eta}_S(Y_1) = \frac{0}{3} = 0$$

$$\bar{\eta}_S(Y_2) = \frac{2}{4} = \frac{1}{2} \qquad \underline{\eta}_S(Y_2) = \frac{1}{2}$$

$$\bar{\eta}_S(Y_3) = \frac{3}{5} \qquad \underline{\eta}_S(Y_3) = \frac{2}{3}$$

$$\bar{\eta}_S(Y_4) = \frac{4}{7} \qquad \underline{\eta}_S(Y_4) = \frac{1}{4}$$

$$\bar{\eta}_S(Y_5) = \frac{5}{7} \qquad \underline{\eta}_S(Y_5) = \frac{4}{5}$$

Wprowadzimy jeszcze pojęcie *dyspersji* zbioru $Y \subset X$ w systemie informacyjnym S, zdefiniowane jako

$$\delta_S(Y) = \frac{\text{card}(\bar{S}Y) - \text{card}(\underline{S}Y)}{\text{card}(Y)} = \frac{\text{card}(B_S(Y))}{\text{card}(Y)}$$

Dyspersja zbiorów rozpatrywanych w poprzednim przykładzie jest następująca:

$$\delta_S(Y_1) = \frac{6}{3} = 2$$

$$\delta_S(Y_2) = \frac{3}{2} = 1{,}5$$

$$\delta_S(Y_3) = \frac{3}{3} = 1$$

$$\delta_S(Y_4) = \frac{6}{4} = 1{,}5$$

$$\delta_S(Y_5) = \frac{3}{5} = 0{,}6$$

Wprowadzone powyżej pojęcia mają prostą interpretację. Dla ich zilustrowania wróćmy do rozpatrywanego na początku przykładu diagnozy medycznej. Jeżeli Y jest zbiorem osób chorych, wyznaczonych przez eksperta na podstawie swej wiedzy i doświadczenia, i chcemy ten zbiór zdefiniować, określając wartości odpowiednich atrybutów w systemie informacyjnym $S = \langle X, A, V, \varrho \rangle$, $Y \subset X$, to jak wiadomo z poprzednich rozważań, zadanie takie może być niewykonalne, gdyż w ten sposób można zdefiniować tylko takie podzbiory zbioru X, które są sumami zbiorów elementarnych systemu S, a więc na podstawie tylko analizy symptomów choroby nie zawsze można odtworzyć diagnozę eksperta. Można tylko odtworzyć ją w ogólnym przypadku z pewnym przybliżeniem (od dołu bądź od góry). Ograniczenie zbioru Y w systemie S, $B_S(Y)$ zawiera te wszystkie elementy, których zaklasyfikowanie może być nieprawidłowe. Dyspersja $\delta_S(Y)$ daje miarę liczbową tej nieprawidłowości. Oczywiście im dyspersja jest bliższa 1, tym lepiej można opisać w systemie S wiedzę eksperta o zbiorze Y. Na przykład gdyby to samo zadanie postawienia diagnozy było wykonywane niezależnie przez eksperta, który stawia nieomylne diagnozy (oczywiście jest to założenie teoretyczne), i studenta, który stawia diagnozę na podstawie występowania bądź nie określonych symptomów, wówczas ograniczenie $B_S(Y)$ oraz dyspersja $\delta_S(Y)$ pokazywałaby różnicę między tymi dwoma sposobami stawiania diagnozy.

Zilustrujemy jeszcze na tym samym przykładzie pojęcia dolnego i górnego przybliżenia oraz ich dokładności.

Jeżeli w badanym zbiorze pacjentów X podzbiór $Y \subset X$ jest zbiorem osób chorych, a wskutek diagnozy na podstawie symptomów otrzymaliśmy jako zbiór osób chorych $\underline{S}Y$, to $\underline{E}_S(Y)$ oznacza zbiór osób chorych, które nie zostały rozpoznane jako chore. Dokładność zatem naszej diagnozy jest $\underline{\eta}_S(Y)$. Na przykład $\underline{\eta}_S(Y) = 0{,}8$ oznacza, że 20% osób chorych nie zostało rozpoznanych jako chore.

Podobnie, jeżeli zamiast zbioru Y osób chorych otrzymaliśmy w wyniku diagnozy na podstawie symptomów zbiór $\bar{S}Y$, to w zbiorze tym znajdują się wszystkie osoby chore w rzeczywistości, ale również i osoby zdrowe, które zostały zakwalifikowane jako chore. W konsekwencji osoby błędnie zakwalifikowane stanowią zbiór $\bar{E}_S(Y)$, a dokładność naszej diagnozy wynosi $\bar{\eta}_S(Y)$. Na przykład $\bar{\eta}_S(Y) = 0{,}75$ oznacza, że 25% osób w zbiorze osób uznanych za chore jest zdrowych.

Podana ilustracja wyjaśnia intuicyjny sens wprowadzanych pojęć. Bardziej szczegółowo zajmiemy się nimi w dalszych punktach.

8.3. WŁASNOŚCI PRZYBLIŻEŃ

Łatwo sprawdzić, że przybliżenia w systemie S mają własności podane w następujących aksjomatach:

A1. $\bar{S}Y \supset Y \supset \underline{S}Y$

A2. $\bar{S}1 = \underline{S}1 = 1$

A3. $\bar{S}0 = \underline{S}0 = 0$

A4. $\bar{S}\bar{S}Y = \underline{S}\bar{S}Y = \bar{S}Y$

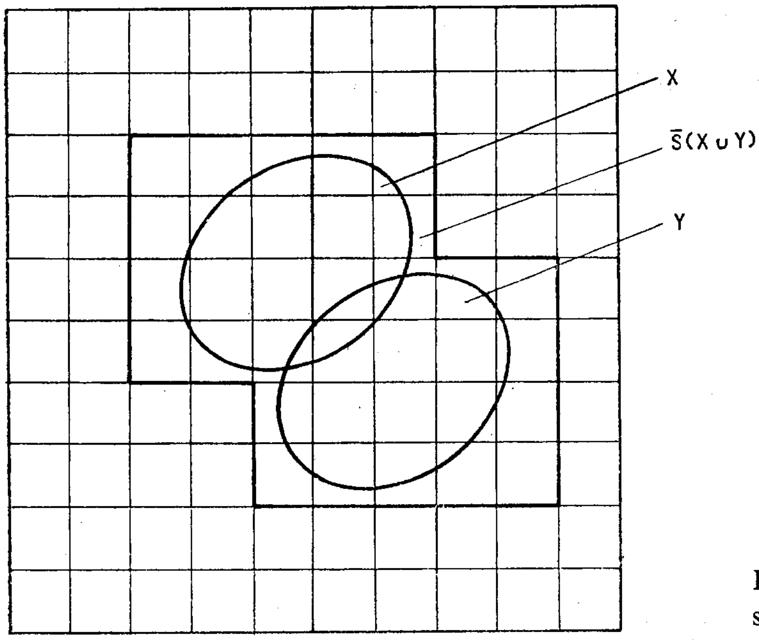
A5. $\underline{S}\underline{S}Y = \bar{S}\underline{S}Y = \underline{S}Y$

A6. $\bar{S}(Y \cup Z) = \bar{S}Y \cup \bar{S}Z$

A7. $\underline{S}(Y \cap Z) = \underline{S}Y \cap \underline{S}Z$

A8. $\bar{S}Y = -\underline{S}(-Y)$

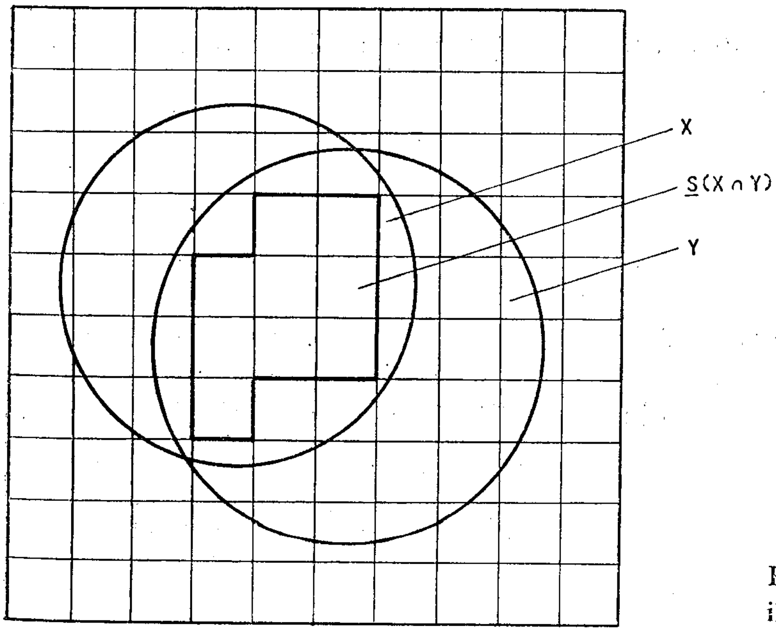
A9. $\underline{S}Y = -\bar{S}(-Y)$



Rys. 8.5. Przybliżenie górne sumy zbiorów

Aksjomaty te są aksjomatami domknięcia i wnętrza topologicznego. Skomentujemy krótko aksjomaty A6÷A9, pozostałe bowiem są oczywiste.

Aksjomat A6 mówi, że przybliżenie górne sumy zbiorów jest sumą górnych przybliżeń zbiorów, aksjomat A7 zaś mówi, że przybliżenie dolne iloczynu zbiorów jest iloczynem dolnych przybliżeń zbiorów. Geometryczną interpretację tych aksjomatów pokazano na rysunkach 8.5, 8.6.



Rys. 8.6. Przybliżenie dolne iloczynu zbiorów

Przykład 8.2

Rozpatrzmy system informacyjny $S = \langle X, A, V, \varrho \rangle$ określony jak w przykładzie 8.1.

X	a	b	c
x_1	u_1	v_1	w_1
x_2	u_2	v_1	w_2
x_3	u_1	v_2	w_2
x_4	u_1	v_1	w_1
x_5	u_2	v_1	w_2
x_6	u_2	v_2	w_1
x_7	u_2	v_1	w_2
x_8	u_1	v_1	w_1

Zbiorami elementarnymi w tym systemie są

$$E_1 = \{x_1, x_4, x_8\}$$

$$E_2 = \{x_2, x_5, x_7\}$$

$$E_3 = \{x_3\}$$

$$E_4 = \{x_6\}$$

Niech

$$Y_1 = \{x_1, x_3, x_5\}$$

$$Y_2 = \{x_2, x_3, x_6\}$$

Na podstawie aksjomatu A6 górne przybliżenie sumy zbiorów $Y_1 \cup Y_2$ jest równe $\bar{S}Y_1 \cup \bar{S}Y_2$, tj.

$$\bar{S}Y_1 = E_1 \cup E_2 \cup E_3$$

oraz

$$\bar{S}Y_2 = E_2 \cup E_3 \cup E_4$$

skąd

$$\bar{S}(Y_1 \cup Y_2) = E_1 \cup E_2 \cup E_3 \cup E_4 = X$$

Na podstawie aksjomatu A7 dolne przybliżenie iloczynu zbiorów $Y_1 \cap Y_2$ jest równe $\underline{S}Y_1 \cap \underline{S}Y_2$, tzn.

$$\underline{S}Y_1 = E_3$$

$$\underline{S}Y_2 = E_3 \cap E_4$$

skąd

$$\underline{S}(Y_1 \cap Y_2) = E_3 \cap (E_3 \cup E_4) = E_3 = \{x_3\}$$

Aksjomaty A8 i A9 wiążą dolne i górne przybliżenia, jak to zilustrowano poniżej:

$$\bar{S}Y_1 = E_1 \cup E_2 \cup E_2$$

Dolne przybliżenie zbioru $-Y_1 = \{x_2, x_4, x_6, x_7, x_8\}$ jest równe

$$\underline{S}(-Y_1) = E_4$$

zaś

$$-\underline{S}(-Y_1) = -E_4 = E_1 \cup E_2 \cup E_3$$

Własność A8 jest więc spełniona.

Podobnie

$$\underline{S}Y_1 = E_3$$

natomiast górne przybliżenie zbioru $-Y_1 = \{x_2, x_4, x_6, x_7, x_8\}$ jest równe

$$\bar{S}(-Y_1) = E_1 \cup E_2 \cup E_3$$

a więc

$$-\bar{S}(-Y_1) = -(E_1 \cup E_2 \cup E_4) = E_3$$

Własność A9 jest więc spełniona. ■

Z aksjomatów A1÷A9 można wyprowadzić następujące własności:

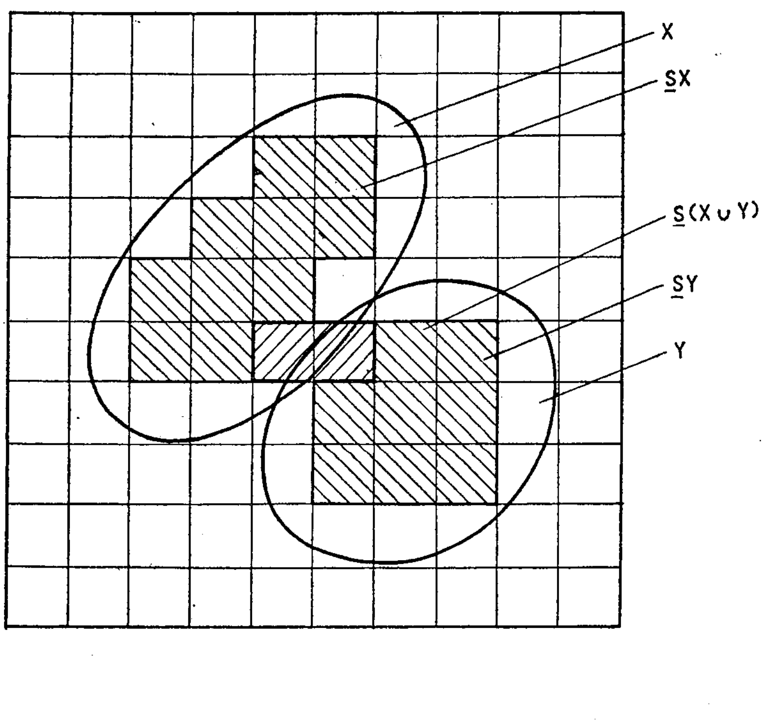
(1) $\bar{S}(Y \cap Z) \subset \bar{S}Y \cap \bar{S}Z$

(2) $\underline{S}(Y \cup Z) \supset \underline{S}Y \cup \underline{S}Z$

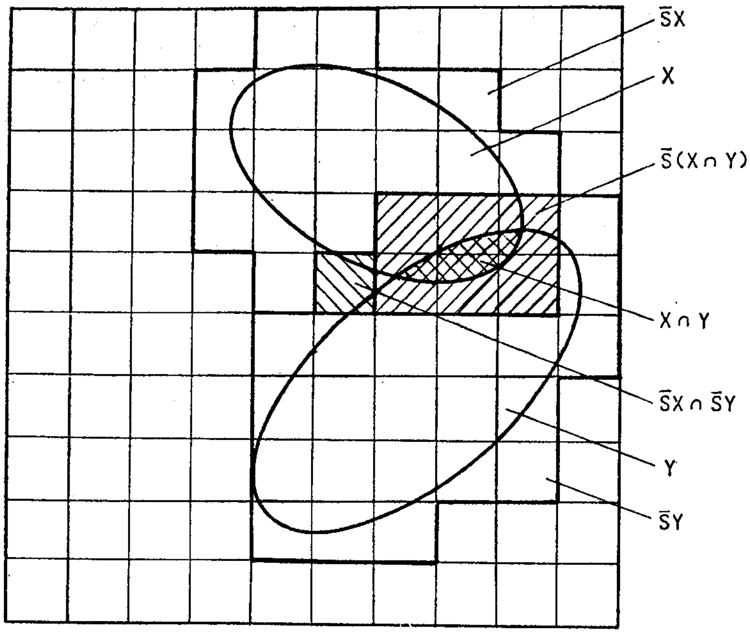
(3) $\bar{S}(Y - Z) \supset \bar{S}Y - \bar{S}Z$

(4) $\underline{S}(Y - Z) \subset \underline{S}Y - \underline{S}Z$

Geometryczną interpretację własności (1) i (2) pokazano na rysunkach 8.7 i 8.8.



Rys. 8.7. Przybliżenie dolne sumy zbiorów



Rys. 8.8. Przybliżenie górne iloczynu zbiorów

Własność (1) mówi, że górne przybliżenie iloczynu zbiorów nie zawsze można zastąpić iloczynem górnych przybliżeń tych zbiorów, własność (2) zaś nie pozwala zastąpić dolnego przybliżenia sumy zbiorów sumą ich dolnych przybliżeń.

Przykład 8.3

Przyjmijmy system informacyjny z przykładu 8.2 oraz rozpatrzmy górne przybliżenie iloczynu zbiorów $Y_1 = \{x_1, x_3, x_5\}$ oraz $Y_2 = \{x_2, x_3, x_4, x_6\}$

$$\bar{S}(Y_1 \cap Y_2) = \bar{S}(\{x_3\}) = E_3$$

Natomiast

$$\bar{S}Y_1 \cap \bar{S}Y_2 = (E_1 \cup E_2 \cup E_3) \cap (E_1 \cup E_2 \cup E_3 \cup E_4) = E_1 \cup E_2 \cup E_3$$

A więc

$$\bar{S}(Y_1 \cap Y_2) \subset \bar{S}Y_1 \cap \bar{S}Y_2$$

gdyż

$$E_3 \subset E_1 \cup E_2 \cup E_3$$

Podobnie

$$\underline{S}(Y_1 \cup Y_2) = \underline{S}(\{x_1, x_2, x_3, x_4, x_5, x_6\}) = E_3 \cup E_4$$

natomiast

$$\underline{S}Y_1 = E_3$$

oraz

$$\underline{S}Y_2 = E_3 \cup E_4$$

a więc

$$\underline{S}Y_1 \cup \underline{S}Y_2 = E_3 \cup E_4$$

Własność (2) jest więc spełniona, gdyż

$$E_3 \cup E_4 \subset E_3 \cup E_4$$

(w tym szczególnym przypadku zachodzi nawet równość). ■

Podane w tym punkcie własności przybliżeń dają podstawową charakterystykę tego pojęcia.

8.4. PRZYBLIŻONY OPIS PODZBIORÓW

Obecnie możemy wrócić do ściślejszego sformułowania problemu postawionego na początku tego rozdziału.

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym, niech $Y \subset X$ oraz niech L_S będzie językiem systemu S.

Powiemy, że $t \in L_S$ jest *przybliżonym opisem zbioru Y od dołu*, gdy $\sigma_S(t) = \underline{S}Y$; $t \in L_S$ jest *przybliżonym opisem zbioru Y od góry*, gdy $\sigma_S(t) = \bar{S}Y$.

Przybliżony opis zbioru Y w systemie S od góry oznaczymy przez $\bar{\tau}_S(Y)$ (lub krótko $\bar{\tau}(Y)$), opis przybliżony od dołu zaś przez $\underline{\tau}_S(Y)$, (lub krótko $\underline{\tau}(Y)$). Mówiąc inaczej, jeżeli $\tau(Y)$ oznacza term t taki, że $\sigma(t) = Y$, to $\underline{\tau}_S(Y) = \tau(\underline{S}Y)$ oraz $\bar{\tau}_S(Y) = \tau(\bar{S}Y)$.

Ponieważ każdy term w języku L_S można sprowadzić do postaci normalnej, będziemy przyjmować, że zarówno przybliżony opis od góry, jak i od dołu dowolnego zbioru Y jest termem w postaci normalnej.

Z definicji opisów przybliżonych wynika, że dla otrzymania przybliżonego opisu zbioru Y od dołu należy wziąć sumę wszystkich termów prostych odpowiadających zbiorom elementarnym wchodzących w skład zbioru $\underline{S}Y$.

Podobnie dla otrzymania przybliżonego opisu zbioru Y od góry należy wziąć sumę wszystkich termów prostych odpowiadającym zbiorom elementarnym wchodzącym w skład zbioru $\bar{S}Y$.

Przykład 8.4

Weźmy pod uwagę system informacyjny rozpatrywany w poprzednim punkcie i określony tablicą

X	a	b	c
x_1	u_1	v_1	w_1
x_2	u_2	v_1	w_2
x_3	u_1	v_2	w_2
x_4	u_1	v_1	w_1
x_5	u_2	v_1	w_2
x_6	u_2	v_2	w_1
x_7	u_2	v_1	w_2
x_8	u_1	v_1	w_1

Zbiorom elementarnym tego systemu

$E_1 = \{x_1, x_4, x_8\}$

$E_2 = \{x_2, x_5, x_7\}$

$E_3 = \{x_3\}$

$E_4 = \{x_6\}$

odpowiadają odpowiednio następujące termy elementarne:

$t_1 = (a, u_1) \cdot (b, v_1) \cdot (c, w_1)$

$t_2 = (a, u_2) \cdot (b, v_1) \cdot (c, w_2)$

$t_3 = (a, u_1) \cdot (b, v_2) \cdot (c, w_2)$

$t_4 = (a, u_2) \cdot (b, v_2) \cdot (c, w_1)$

Rozpatrzymy przybliżone od góry i od dołu opisy następujących zbiorów:

$Y_1 = \{x_1, x_4, x_5\}$

$Y_2 = \{x_3, x_5\}$

$Y_3 = \{x_6, x_3, x_8\}$

$Y_4 = \{x_1, x_4, x_5, x_6\}$

$Y_5 = \{x_1, x_2, x_3, x_5, x_7\}$

Ponieważ przybliżenia tych zbiorów są następujące:

$\bar{S}Y_1 = E_1 \cup E_2$ $\quad \underline{S}Y_1 = \emptyset$

$\bar{S}Y_2 = E_2 \cup E_3$ $\quad \underline{S}Y_2 = E_3$

$\bar{S}Y_3 = E_1 \cup E_3 \cup E_4$ $\quad \underline{S}Y_3 = E_3 \cup E_4$

$\bar{S}Y_4 = E_1 \cup E_2 \cup E_4$ $\quad \underline{S}Y_4 = E_4$

$\bar{S}Y_5 = E_1 \cup E_2 \cup E_3$ $\quad \underline{S}Y_5 = E_2 \cup E_3$

więc ich przybliżone opisy będą miały postać:

$$\bar{\tau}(Y_1) = (a, u_1) \cdot (b, v_1) \cdot (c, w_1) + (a, u_2) \cdot (b, v_1) \cdot (c, w_2)$$

$$\bar{\tau}(Y_2) = (a, u_2) \cdot (b, v_1) \cdot (c, w_2) + (a, u_1) \cdot (b, v_2) \cdot (c, w_2)$$

$$\bar{\tau}(Y_3) = (a, u_1) \cdot (b, v_1) \cdot (c, w_1) + (a, u_1) \cdot (b, v_2) \cdot (c, w_2) + \\ + (a, u_2) \cdot (b, v_2) \cdot (c, w_1)$$

$$\bar{\tau}(Y_4) = (a, u_1) \cdot (b, v_1) \cdot (c, w_1) + (a, u_2) \cdot (b, v_1) \cdot (c, w_2) + \\ + (a, u_2) \cdot (b, v_2) \cdot (c, w_2)$$

$$\bar{\tau}(Y_5) = (a, u_1) \cdot (b, v_1) \cdot (c, w_1) + (a, u_2) \cdot (b, v_1) \cdot (c, w_2) + \\ + (a, u_1) \cdot (b, v_2) \cdot (c, w_2)$$

$$\underline{\tau}(Y_1) = 0$$

$$\underline{\tau}(Y_2) = (a, u_1) \cdot (b, v_2) \cdot (c, w_2)$$

$$\underline{\tau}(Y_3) = (a, u_1) \cdot (b, v_2) \cdot (c, w_2) + (a, u_2) \cdot (b, v_2) \cdot (c, w_2)$$

$$\underline{\tau}(Y_4) = (a, u_2) \cdot (b, v_2) \cdot (c, w_1)$$

$$\underline{\tau}(Y_5) = (a, u_2) \cdot (b, v_1) \cdot (c, w_2) + (a, u_1) \cdot (b, v_2) \cdot (c, w_2)$$

Tak więc można w sposób przybliżony (od góry bądź od dołu) opisać dowolny podzbiór zbioru obiektów systemu informacyjnego. Wracając ponownie do naszego przykładu wyjściowego, gdy mamy zbiór osób, o których wiemy, że są chore na określoną chorobę, ponadto mamy ustalony zbiór objawów, które możemy sprawdzić u każdej osoby, możemy teraz rozstrzygnąć, czy za pomocą tych objawów możemy jednoznacznie scharakteryzować daną chorobę, czy też nie. Jeżeli nie — możemy podać przybliżony opis choroby (od góry bądź od dołu), który jednak nie gwarantuje ścisłej diagnozy. Możemy natomiast obliczyć dokładność opisów, posługując się wprowadzonymi poprzednio współczynnikami dokładności $\bar{\eta}, \underline{\eta}$ oraz rozproszenia δ.

Interesujące jest pytanie, czy wszystkie atrybuty występujące w systemie są konieczne do opisu zadanego zbioru obiektów. Na przykład stosując aksjomaty podane w drugim rozdziale można wyrażenie

$$\underline{\tau}(Y_3) = (a, u_1) \cdot (b, v_2) \cdot (c, w_2) + (a, u_2) \cdot (b, v_2) \cdot (c, w_2)$$

sprowadzić do postaci:

$$\underline{\tau}(Y_3) = (b, v_2) \cdot (c, w_2)$$

co oznacza, że do przybliżonego opisu od dołu zbioru Y_3 nie jest potrzebny atrybut a

Zagadnienie usuwania zbędnych atrybutów w przybliżonych opisach jest bardzo ważne z punktu widzenia praktycznego, pozwala ono bowiem klasyfikować obiekty na podstawie mniejszej liczby danych. W przypadku zastosowań medycznych ma to szczególnie ważne znaczenie, gdyż pozwala postawić diagnozę przybliżoną na podstawie obserwacji mniejszej liczby objawów.

Możemy tu pytać o minimalny zbiór atrybutów potrzebny do opisania danego zbioru Y w systemie S. Zbiorów takich może być oczywiście w ogólnym przypadku więcej niż jeden.

Zauważmy związek poruszanego tu zagadnienia z minimalizacją zbioru atrybutów w systemie informacyjnym rozpatrywanym w rozdz. 1. Minimalizacja atrybutów w systemie informacyjnym pozwalała usunąć wszystkie te atrybuty, które nie miały istotnego wpływu na relacje równoważności generowaną przez system informacyjny, tj. na podział zbioru obiektów na zbiory elementarne.

Minimalizacja atrybutów w opisie przybliżonym zbioru obiektów systemu może być różna dla różnych zbiorów obiektów. Znaczy to np., że dla każdej choroby możemy mieć specyficzny dla niej zbiór objawów, wystarczający dla jej identyfikacji (ewentualnie z przybliżeniem). Jest to więc całkowicie zgodne z potoczną intuicją.

Jakkolwiek problem ten jest bardzo ważny, nie będziemy się tu nim zajmować, gdyż można go rozwiązać standardowymi metodami algebry Boole'a stosowanymi np. w uproszczeniach sieci przełączających (patrz np. praca Imielińskiego [20]).

8.5. PRÓBKA ZBIORU

W dotychczasowych rozważaniach przyjmowaliśmy, że dla znaleziemia przybliżonego opisu (od góry bądź od dołu) zbioru Y w systemie informacyjnym S, tj. dla znalezienia termu $\bar{\tau}_S(Y)$ bądź $\underline{\tau}_S(Y)$ jest konieczna analiza wszystkich elementów zbioru Y. Powstaje pytanie, czy opisów zbioru Y nie można otrzymać na podstawie analizy części jego elementów. Faktycznie w praktyce taka sytuacja ma największe znaczenie. Na przykład w medycynie, gdy chcemy uzyskać opis jakiejś choroby na podstawie analizy znanych jej przypadków, staramy się znaleźć objawy charakterystyczne danej choroby, ważne oczywiście nie tylko dla rozpatrywanych przypadków, lecz także dla wszystkich innych.

Powstaje tutaj pytanie, kiedy na podstawie podzbioru zbioru Y można znaleźć opis zbioru Y?

Obecnie sformułujemy problem ten nieco ściślej.

Niech $S=\langle X, A, V, \varrho\rangle$ będzie systemem informacyjnym oraz niech $Y' \subset Y$ oraz $Y \subset X$.

Powiemy, że Y' jest *próbką* zbioru Y w systemie informacyjnym S, gdy

(1) $\bar{\tau}_S(Y)=\bar{\tau}_S(Y')$

lub

(2) $\underline{\tau}_S(Y)=\bar{\tau}_S(Y')$

Gdy jest spełniony warunek (1), Y' będziemy nazywać *próbką górną*, gdy zaś jest spełniony warunek (2), Y' będziemy nazywać *próbką dolną*.

Jeżeli Y' jest próbką górną (dolną) Y w systemie S oraz nie istnieje próbka górna (dolna) Y'' zbioru Y w systemie S, taka że $Y' \subset Y''$, to Y' nazwiemy *minimalną próbką górną (dolną)* Y w systemie S.

Łatwo zauważyć, że Y' jest minimalną próbką górną (dolną) Y w systemie S wtedy i tylko wtedy, gdy Y' zawiera dokładnie po jednym elemencie z każdego zbioru elementarnego wchodzącego w skład zbioru $\bar{S}Y(\underline{S}Y)$.

Powyższa własność wynika bezpośrednio z definicji próbki oraz przybliżenia górnego i dolnego.

Przykład 8.5

Rozpatrzmy system informacyjny, taki jak w poprzednich punktach, o następującej tabelce:

X	a	b	c
x_1	u_1	v_1	w_1
x_2	u_2	v_1	w_2
x_3	u_1	v_2	w_2
x_4	u_1	v_1	w_1
x_5	u_2	v_1	w_2
x_6	u_2	v_2	w_1
x_7	u_2	v_1	w_2
x_8	u_1	v_1	w_1

posiadający zbiory elementarne

$$E_1 = \{x_1, x_4, x_8\}$$
$$E_2 = \{x_2, x_5, x_7\}$$
$$E_3 = \{x_3\}$$
$$E_4 = \{x_6\}$$

Niech $Y_1 = \{x_1, x_4, x_5\}$. Ponieważ $\bar{S}Y_1 = E_1 \cup E_2$, więc górnymi próbkami minimalnymi zbioru Y_1 są np. zbiory $\{x_1, x_2\}, \{x_1, x_5\}, \{x_4, x_7\}, \{x_8, x_2\}$ itd.

Dolna próbka minimalna zbioru Y_1 jest zbiorem pustym, gdyż $\underline{S}Y_1 = \emptyset$.

Dla zbioru $Y_5 = \{x_1, x_2, x_3, x_5, x_7\}$ mamy $\bar{S}Y_5 = E_1 \cup E_2 \cup E_3$ oraz $\underline{S}Y_5 = E_2 \cup E_3$. Minimalnymi próbkami górnymi zbioru Y_5 są np. zbiory $\{x_1, x_2, x_3\}$, $\{x_4, x_7, x_3\}, \{x_1, x_7, x_3\}$ itd. Minimalnymi próbkami dolnymi zbioru Y_5 są na przykład zbiory $\{x_2, x_3\}, \{x_7, x_3\}$. ■

Podane tu przykłady mają charakter elementarny i nie widać z nich celowości znajdowania opisu na podstawie próbek zamiast na podstawie zbiorów właściwych. W rzeczywistości zbiory, których opisu poszukujemy, mogą być bardzo duże, ich próbki minimalne zaś niewielkie, tak że celowość użycia próbki zamiast zbioru właściwego jest wtedy oczywista.

Podajmy na zakończenie, że znalezienie minimalnej próbki może być w praktyce sprawą kłopotliwą. Wybranie przypadkowe elementów próbki nie może dać pożądanego rezultatu. W szczególności jeżeli liczność zbioru Y' jest mniejsza od liczby zbiorów elementarnych wchodzących w skład odpowiedniego przybliżenia zbioru Y, to Y' na pewno nie może być próbką zbioru Y.

8.6. PRZYKŁAD ZASTOSOWANIA

Rozpatrzmy bardzo prosty przykład z medycyny ilustrujący nasze dotychczasowe wywody.

Będziemy rozpatrywali niektóre objawy występujące w chorobach tarczycy, według danych uzyskanych od prof. Doroszewskiego [114]. Dla uproszczenia będziemy rozpatrywali tylko następujące objawy:

(a) ogólne osłabienie,
(b) stany podgorączkowe,
(c) wzmożona pobudliwość nerwowa,
(d) nadmierna potliwość,
(e) wzmożone pragnienie,
(f) obrzęk powiek,
(g) bicie serca,
(h) bóle serca,
(i) zaburzenie miesiączkowania.

Objawy te będziemy traktować jako atrybuty. Przyjmiemy, że każdy z powyższych atrybutów może przyjmować jedną z wartości 0, 1. Zero oznacza, że dany objaw nie występuje, jeden zaś oznacza występowanie odpowiedniego objawu.

Dla uproszczenia zamiast atrybutów będziemy w dalszym ciągu posługiwać się oznaczającymi je literami.

Termami prostymi w języku tego systemu będą wyrażenia, takie jak np.

$$(a,0)\cdot(b,1)\cdot(c,0)\cdot(d,1)\cdot(e,0)\cdot(f,1)\cdot(g,0)\cdot(\mathrm{h},0)\cdot(i,1)$$

które oznacza, że nie występuje ogólne osłabienie, występują stany podgorączkowe, nie występuje wzmożona pobudliwość nerwowa itd.

Termy elementarne można przedstawiać w prostej postaci, jako ciągu zer i jedynek, których pozycjom odpowiadają kolejne atrybuty. Tak więc poprzedni term zapiszemy w postaci

0 1 0 1 0 1 0 0 1

Oczywiście w systemie tym istnieje $2^9 = 512$ różnych termów elementarnych. Przyjmijmy, że dla określonej grupy badanych pacjentów otrzymano jedynie następujące niepuste termy elementarne:

$$t_1 = 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1$$

$$t_2 = 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1$$

$$t_3 = 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0$$

$$t_4 = 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0$$

$$t_5 = 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0$$

Znaczy to, że badani pacjenci zostali w ten sposób podzieleni na pięć rozłącznych grup, które stanowią zbiory elementarne naszego systemu. Przyjmijmy ponadto, że na podstawie szczegółowych badań i obserwacji stwierdzono, że do grup 1 i 2 zalicza się pacjentów, u których stwierdzono, że na pewno nie mają choroby tarczycy, do grup 4 i 5 pacjentów, u których stwierdzono, że na pewno mają chorobę tarczycy, do grupy 3 zaś należą pacjenci, u których nie udało się ani stwierdzić z całą pewnością, ani wykluczyć choroby tarczycy. Dalsze badania mogą dopiero ewentualnie rozstrzygnąć, czy pacjenci są zdrowi czy chorzy.

Powstaje pytanie, jak dalece podane tu objawy można uznać na podstawie posiadanego materiału za charakterystyczne dla choroby tarczycy? Na pytanie to nie potrafimy dać dokładnej odpowiedzi, możemy dać jedynie opisy choroby, stanowiące przybliżenia od góry bądź od dołu. Przybliżenie dolne zbioru chorych będzie opisane termem t_4+t_5, przybliżenie górne zaś termem $t_3+t_4+t_5$. Podobnie przybliżenie dolne ludzi zdrowych jest opisane termem t_1+t_2, przybliżenie górne ludzi zdrowych zaś jest opisane przez term $t_1+t_2+t_3$.

Łatwo zauważyć, że w termie t_4+t_5 można wyeliminować trzeci atrybut, wyciągając wszystkie pozostałe atrybuty przed nawias i zastępując zgodnie z aksjomatami algebry Boole'a wyrażenie $(c+\sim c)$ jedynką, otrzymując w rezultacie

$$t_4+t_5=(\sim a)bde(\sim f)gh(\sim i)$$

Zauważmy, że dla odróżnienia pacjentów, o których wiemy, że nie są chorzy na tarczycę (tj. należących do grup 1 lub 2), od pacjentów, o których wiemy, że są chorzy na tarczycę (grupa 9 i 5) nie są potrzebne atrybuty a i d, gdyż przyjmują one jednakowe wartości dla obu grup.

8.7. KLASYFIKACJA WIELOWARTOŚCIOWA

W dotychczasowych rozważaniach zajmowaliśmy się klasyfikacją dwuwartościową, tj. interesowaliśmy się, czy badany obiekt należy do określonego zbioru Y bądź też — jego uzupełnienia. Klasyfikowaliśmy więc obiekty na dwie grupy, np. zdrowych i chorych.

Często jednak zachodzi konieczność klasyfikowania przedmiotu do jednej z n grup, gdzie $n>2$. Klasyfikację taką będziemy nazywać wielowartościową. Przeniesiemy obecnie nasze dotychczasowe rozważania na klasyfikację wielowartościową.

Niech $S=\langle X, A, V, \varrho\rangle$ będzie systemem informacyjnym i niech $C(X)=\{X_1, X_2, \ldots, X_n\}$ będzie podziałem zbioru X, tj. $X_i \subset X$, $X_i \cap X_j=\emptyset$ dla $i \neq j$ oraz $\bigcup_{i=1} X_i=X$. Zbiory X_i nazywamy *blokami podziału* $C(X)$.

Jeżeli istnieje takie X_i, że X_i jest zbiorem nieopisywalnym w systemie informacyjnym S, to podział $C(X)=\{X_1, X_2, \ldots, X_n\}$ nazwiemy *nieopisywalnym* w systemie S.

Dolnym przybliżeniem podziału $C(X)=\{X_1, X_2, \ldots, X_n\}$ w systemie informacyjnym S będziemy nazywali rodzinę zbiorów

$$\underline{C}_S(X)=\{Y_1, Y_2, \ldots, Y_n\}$$

taką, że $Y_i \subset X$ oraz $Y_i=\underline{S}X_i$ dla każdego $i=1, 2, \ldots, n$.

Podobnie *górnym przybliżeniem podziału* $C(X)=\{X_1, X_2, \ldots, X_n\}$ w systemie S nazwiemy rodzinę zbiorów $\bar{C}_S(X)=\{Y_1, Y_2, \ldots, Y_n\}$, taką, że $Y_i \subset X$ oraz $Y_i=\bar{S}X_i$ dla każdego $i=1, 2, \ldots, n$.

Jeżeli Y jest zbiorem elementarnym systemu $S=\langle X, A, V, \varrho\rangle$ oraz $Y\subset\bigcup_{i=1}^{n}\underline{S}X_i$, to Y nazwiemy *wewnętrznym* zbiorem elementarnym podziału $C(X)=\{X_1, X_2, \ldots, X_n\}$. Jeżeli $Y\cap\bigcup_{i=1}^{n}\underline{S}X_i=\emptyset$, to Y nazwiemy *brzegowym* zbiorem elementarnym podziału $C(X)=\{X_1, X_2, \ldots, X_n\}$.

Zauważmy, że dla dolnego przybliżenia podziału $C(X)=\{X_1, X_2, \ldots, X_n\}$:

(1) $\underline{S}X_i\cap\underline{S}X_j=\emptyset$ dla każdego i, j $i\neq j$

(2) $\bigcup_{i=1}^{n}\underline{S}X_i\subset X$

zaś dla górnego przybliżenia podziału $C(X)$, gdy $C(X)$ nie składa się z samych zbiorów opisywalnych:

(3) $\bar{S}X_i\cap\bar{S}X_j\neq\emptyset$ dla pewnych i, j $i\neq j$

(4) $\bigcup_{i=1}^{n}\bar{S}X_i=X$

Tak więc ani górne, ani dolne przybliżenie dowolnego podziału nieopisywalnego w systemie S nie jest podziałem.

W przybliżeniu dolnym podziału nieopisywalnego suma dolnych przybliżeń bloków nie daje całego zbioru X, natomiast w przybliżeniu górnym, górne przybliżenia bloków podziału mogą mieć niepustą część wspólną. Przybliżenie górne podziału jest więc pokryciem zbioru X, a definiowana przez nie relacja jest nazywana *relacją tolerancji* [116].

Przybliżenie górne podziału $C(X)=\{X_1, X_2, \ldots, X_n\}$ w dowolnym systemie informacyjnym definiuje więc relację binarną $\bar{R}\subset X\times X$ spełniającą następujące warunki:

(5) $\bar{R}(x, x)$ dla każdego $x\in X$,

(6) Jeżeli $\bar{R}(x, y)$, to $\bar{R}(y, x)$.

Przybliżenie dolne podziału $C(X)$ definiuje relacje $\underline{R}\subset X\times X$ spełniającą warunki:

(7) Jeżeli $\underline{R}(x, y)$, to $\underline{R}(y, x)$,

(8) Jeżeli $\underline{R}(x, y)$ oraz $\underline{R}(y, z)$, to $\underline{R}(x, z)$.

Relacja $\bar{R}$ nie jest przechodnia, relacja $\underline{R}$ zaś nie jest zwrotna. Obie relacje $\bar{R}, \underline{R}$ można traktować jako przybliżone relacje równoważności.

Problem opisu bloków podziału wielowartościowego jest dokładnie taki sam jak w przypadku podziału dwuwartościowego. Nie będziemy się nim tu zajmować.

8.8. DOKŁADNOŚĆ PRZYBLIŻONEJ KLASYFIKACJI

Podobnie jak to uczyniliśmy dla określenia dokładności przybliżeń zbioru Y w systemie informacyjnym S, możemy wprowadzić współczynniki określające dokładność przybliżenia podziału $C(X)$ w systemie S.

Niech $C(X)=\{X_1, X_2, \ldots, X_n\}$ będzie podziałem zbioru X i niech $S=\langle X, A, V, \varrho\rangle$ będzie systemem informacyjnym. Dokładność podziału $C(X)$ w systemie informacyjnym S oznaczymy przez $\alpha_S C(X)$ i zdefiniujemy następująco:

$$\alpha_S C(X)=\frac{\sum_{i=1}^{n} \operatorname{card} \underline{S}(X_i)}{\operatorname{card}(X)}$$

Współczynnik ten wyraża stosunek największej możliwej liczby obiektów trafnie zakwalifikowanych do bloków podziału $C(X)$ za pomocą atrybutów systemu S — do liczby wszystkich obiektów w systemie S. Oczywiście współczynnik ten jest zawarty między zerem a jednością.

W przypadku klasyfikacji dwuwartościowej (tj. podziału zbioru X na dwa bloki Y i $-Y$) współczynnik ten przyjmie postać

$$\alpha_S C(X)=\frac{\operatorname{card}(\underline{S}Y)+\operatorname{card}(\underline{S}(-Y))}{\operatorname{card}(X)}$$

Jeżeli $C(X)=\{X_1, X_2, \ldots, X_n\}$ oraz $S=\langle X, A, V, \varrho\rangle$, to zbiór

$$N_S C(X)=X-\bigcup_{i=1}^{n} \underline{S}X_i=\bigcup_{i=1}^{n} B_S(X_i)$$

oznacza zbiór obiektów, które mogą być niewłaściwie zaklasyfikowane za pomocą atrybutów systemu S w podziale $C(X)$, natomiast

$$\mu_S C(X)=\frac{\operatorname{card}(N_S C(X))}{\operatorname{card}(X)}$$

oznacza stosunek liczby obiektów, które mogą być niewłaściwie zaklasyfikowane do liczby wszystkich obiektów.

Współczynnik $\mu_S C(X)$ nazwiemy *współczynnikiem dyspersji* podziału $C(X)$ w systemie informacyjnym S.

Oczywiście

$$\mu_S C(X)=1-\alpha_S C(X)$$

Współczynniki $\alpha_S C(X)$ oraz $\mu_S C(X)$ określają jak dokładnie podział $C(X)$ może być opisany za pomocą atrybutów w systemie informacyjnym S.

8.9. UWAGI KOŃCOWE

We wszystkich rozpatrywanych do tej pory przypadkach rozważaliśmy następującą sytuację:

Dany był podział pewnego zbioru i podział ten chcieliśmy opisać za pomocą własności obiektów. Okazało się, że zadanie to nie zawsze może być wykonane. W ogólnym przypadku możliwe jest tylko przybliżone opisanie zadanego podziału za pomocą własności obiektów. Jeżeli zadany podział traktować jako odzwierciedlenie wiedzy jakiegoś eksperta, który podziału tego dokonał, to wiedzy tej w ogólnym przypadku nie da się odtworzyć na podstawie jedynie własności obiektów, wyrażonych poprzez atrybuty.

Badania teoretyczne nad problemem przedstawiania wiedzy są ostatnio prowadzone bardzo szeroko, w związku z różnymi nowymi zastosowaniami komputerów (patrz np. praca [14]).

Przedstawiona tu propozycja różni się nieco od badań prowadzonych w tym zakresie przez wprowadzenie klasyfikacji przybliżonej za pomocą zbiorów elementarnych.

9. NIEPEŁNA KLASYFIKACJA OBIEKTÓW

9.1. UWAGI WSTĘPNE

W dotychczasowych rozważaniach przyjmowaliśmy, że ekspert potrafi zaklasyfikować każdy obiekt prawidłowo, tj. że jest on nieomylny oraz ma dostateczną wiedzę, aby zaklasyfikować obiekt. Warunki te nie zawsze muszą być spełnione. Może się zdarzyć, że ekspert popełni błąd w klasyfikacji, może się też zdarzyć, że nie potrafi on niektórych przedmiotów zaklasyfikować. Na przykład lekarze w niektórych przypadkach mogą nie potrafić stwierdzić, czy pacjent jest chory, czy też nie, na określoną chorobę.

W rozdziale tym zajmiemy się bliższym zbadaniem tego rodzaju przypadków.

9.2. NIEPEŁNA KLASYFIKACJA DWUWARTOŚCIOWA

Przypuśćmy, że ekspert dokonujący klasyfikacji obiektów na dwie klasy Y, $-Y$ nie potrafi w niektórych przypadkach rozstrzygnąć, do której klasy należy badany obiekt. Można wtedy utworzyć trzecią klasę obiektów niezidentyfikowanych. Znaczy to, że zbiór obiektów X dzieli się na trzy klasy X_1, X_2, X_3, otrzymując w ten sposób klasyfikację trójwartościową, w której $Y \subset X_1 \cup X_2$, $-Y \subset X_2 \cup X_3$ oraz $X_1 \subset Y$, $X_3 \subset -Y$. Zbiór X_2 zawiera więc wszystkie elementy, o których ekspert nie potrafi orzec, czy należą one do Y czy do $-Y$.

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym, Y podzbiorem zbioru X oraz $C(X) = \{X_1, X_2, X_3\}$ niech będzie podziałem spełniającym warunki:

(1) $Y \subset X_1 \cup X_2, \qquad -Y \subset X_2 \cup X_3$

(2) $X_1 \subset Y, \qquad X_3 \subset -Y$

Najlepszym przybliżeniem dolnym zbioru Y w systemie S jest wtedy $\underline{S}X_1$, przybliżeniem górnym zbioru Y jest $\bar{S}(X_1 \cup X_2)$. Podobnie dla zbioru $-Y$, przybliżeniem dolnym jest $\underline{S}X_3$, przybliżeniem górnym zaś jest $\bar{S}(X_2 \cup X_3)$.

Tak więc podział dwuwartościowy $C(X) = \{Y, -Y\}$, $Y \subset X$ możemy w takiej sytuacji najlepiej przybliżyć od dołu parą zbiorów

$$\underline{S}X_1, \underline{S}X_3$$

zaś od góry parą zbiorów

$$\bar{S}(X_1 \cup X_2), \bar{S}(X_2 \cup X_3)$$

Odpowiadające im termy

$$\underline{\tau}_S(X_1),\ \underline{\tau}_S(X_3)$$

oraz

$$\underline{\tau}_S(X_1 \cup X_2),\ \bar{\tau}_S(X_2 \cup X_3)$$

stanowią opisy przybliżone podziału dwuwartościowego $C(X)$.

Dokładność przybliżenia górnego i dolnego podziału $C(X)$ może być w tym przypadku gorsza niż w przypadku, gdy ekspert potrafi właściwie zaklasyfikować każdy obiekt, gdyż

$$(3)\quad \frac{\operatorname{card}(\underline{S}X_1)+\operatorname{card}(\underline{S}X_3)}{\operatorname{card}(X)} \leqslant \frac{\operatorname{card}(\underline{S}Y)+\operatorname{card}(\underline{S}(-Y))}{\operatorname{card}(X)}$$

oraz

$$(4)\quad \frac{\operatorname{card}(X)}{\operatorname{card}(\bar{S}(X_1 \cup X_2))+\operatorname{card}(\bar{S}(X_2 \cup X_3))} \leqslant \frac{\operatorname{card}(X)}{\operatorname{card}(\bar{S}Y)+\operatorname{card}(\bar{S}(-Y))}$$

Tak więc niewiedza eksperta może pogorszyć możliwość prawidłowego zaklasyfikowania przedmiotów na podstawie ich własności. Jednakże pogorszenie takie nie zawsze musi wystąpić.

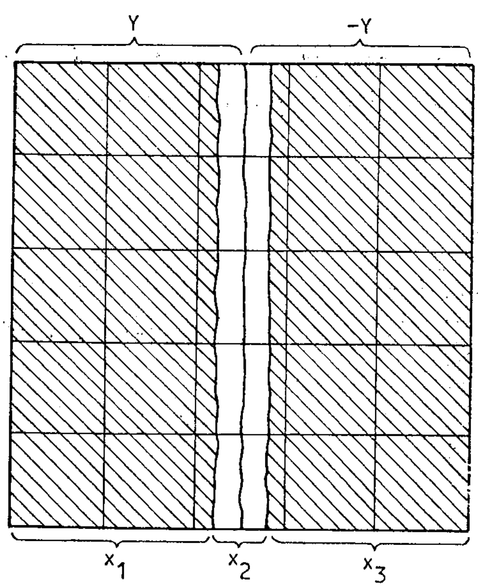
Rozpatrzmy sytuację przedstawioną na rys. 9.1. Siatka przedstawia zbiory elementarne systemu S. Widzimy, że w takim przypadku:

(5) $\underline{S}X_1 = \underline{S}Y$

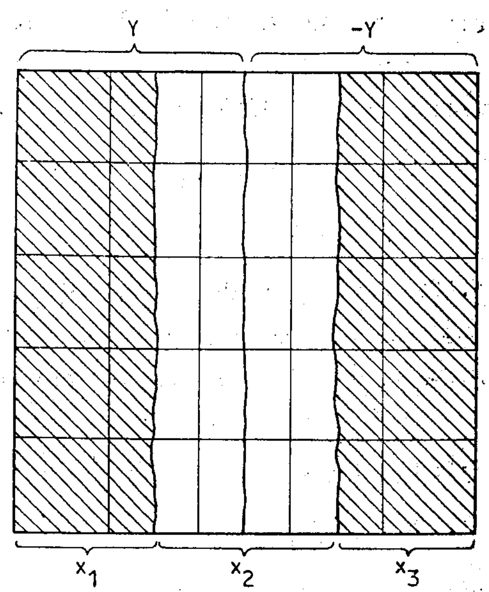
(6) $\underline{S}X_3 = \underline{S}(-Y)$

(7) $\bar{S}(X_1 \cup X_2) = \bar{S}Y$

(8) $\bar{S}(X_2 \cup X_3) = \bar{S}(-Y)$



Rys. 9.1. Przybliżona klasyfikacja (przypadek 1)



Rys. 9.2. Przybliżona klasyfikacja (przypadek 2)

A więc we wzorach (3) i (4) zachodzi równość. Dokładność górnego i dolnego przybliżenia podziału $C(X)$ jest niezależna od wiedzy eksperta. Może on więc w podanych granicach nie potrafić sklasyfikować obiektów, mimo to nie popsuje to dokładności opisu przybliżonego obiektów.

Przypadek taki nie zachodzi, gdy mamy do czynienia z sytuacją pokazaną na rys. 9.2, własności (5), (6), (7) i (8) bowiem przyjmują wtedy postać następującą:

(9) $\underline{S}X_1 \subset \underline{S}Y$

(10) $\underline{S}X_3 \subset \underline{S}(-Y)$

(11) $\bar{S}(X_1 \cup X_2) \supset \bar{S}Y$

(12) $\bar{S}(X_2 \cup X_3) \supset \bar{S}(-Y)$

co w konsekwencji powoduje, że we wzorach (3) i (4) zamiast równości zachodzą nierówności.

Dla ściślejszego sformułowania tego problemu wprowadzimy w następnym punkcie pojęcie zbioru przybliżonego, które pozwoli nam na dokładne rozważenie tej i podobnych sytuacji.

Na zakończenie tego punktu zauważmy, że interesujące są również następujące przybliżenia:

$\bar{S}X_1, \bar{S}X_3$

$\underline{S}X_2, \underline{S}X_2$

oraz odpowiadające im termy. Ich interpretację pozostawimy Czytelnikowi.

9.3. PRZYBLIŻONA RÓWNOŚĆ ZBIORÓW

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym oraz niech $Y, Z \subset X$.

Wprowadzimy trzy rodzaje przybliżonej równości zbiorów w systemie informacyjnym S.

Zbiory Y, Z są *w przybliżeniu równe od dołu* w systemie S, gdy $\underline{S}Y = \underline{S}Z$ (symbolicznie $Y \underset{S}{\bar{\sim}} Z$).

Zbiory Y, Z są *w przybliżeniu równe od góry* w systemie S (symbolicznie $Y \underset{S}{\simeq} Z$), gdy $\bar{S}Y = \bar{S}Z$.

Zbiory Y, Z są *w przybliżeniu równe* w systemie S (symbolicznie $Y \underset{S}{\approx} Z$), gdy

$$Y \underset{S}{\bar{\sim}} Z \quad \text{oraz} \quad Y \underset{S}{\simeq} Z$$

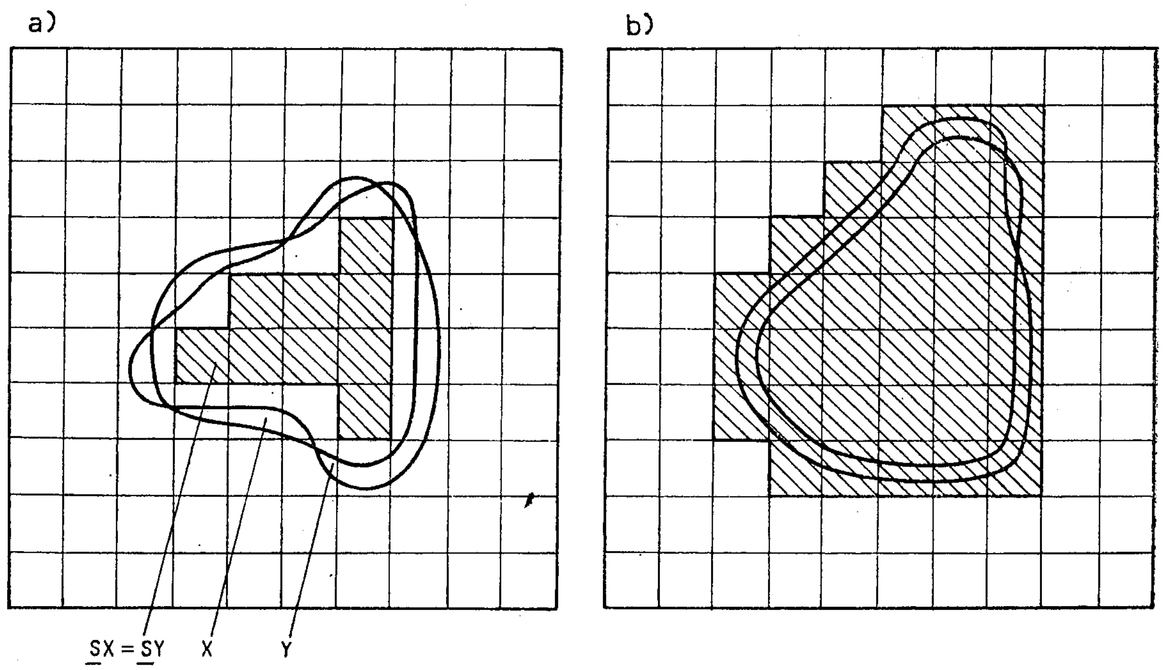
Gdy system S jest ustalony, nie będziemy pisać wskaźnika S przy odpowiednim symbolu przybliżonej równości, pisząc np. $\approx$ zamiast $\underset{S}{\approx}$.

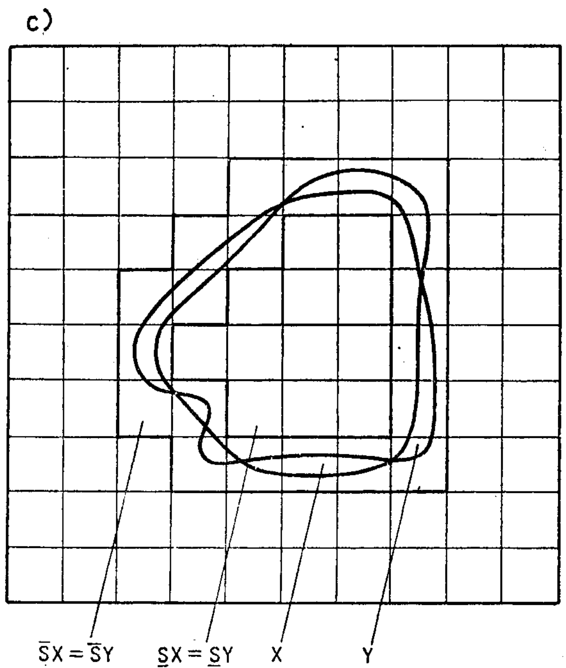
Wszystkie trzy relacje przybliżonej równości zbiorów są relacjami równoważności. Na przykład:

(a) $X \underset{S}{\approx} X$, gdyż $\underline{S}X = \underline{S}X$;

(b) Jeżeli $X \underset{S}{\approx} Y$, to $Y \underset{S}{\approx} X$, gdyż $\underline{S}X = \underline{S}Y$, $\underline{S}Y = \underline{S}X$

(c) Jeżeli $X \underset{S}{\approx} Y$ oraz $Y \underset{S}{\approx} Z$, to $X \underset{S}{\approx} Z$, gdyż jeżeli $\underline{S}X = \underline{S}Y$ oraz $\underline{S}Y = \underline{S}Z$, to $\underline{S}X = \underline{S}Z$.



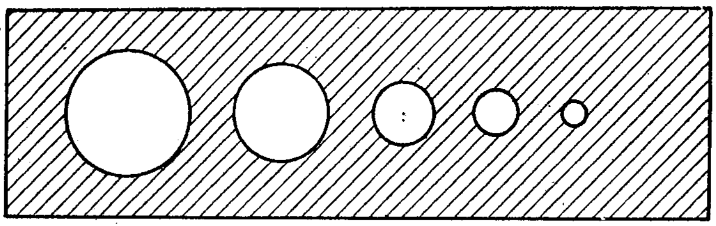


Rys. 9.3: (a) Zbiory w przybliżeniu równe od dołu; (b) Zbiory w przybliżeniu równe od góry; (c) Zbiory w przybliżeniu równe

Pojęciami przybliżonej równości możemy się zatem posługiwać podobnie jak zwykłą równością.

Pojęcie zbioru przybliżonego ilustruje rys. 9.3. Pokazano na nim kolejno zbiory w przybliżeniu równe od dołu, od góry i zbiory w przybliżeniu równe od dołu i góry jednocześnie (zbiory w przybliżeniu równe).

Pojęcie to dobrze odpowiada intuicji technicznej. Przyjmijmy jako system informacyjny układ otworów pomiarowych, jak to pokazano na rys. 9.4. (Dlaczego układ taki można traktować jako system informacyjny?).



Rys. 9.4. Otwory pomiarowe

Dwa wałki mają w przybliżeniu od góry jednakowe średnice w systemie pomiarowym S, jeżeli maksymalne otwory pomiarowe, do których mieszczą się oba wałki są równe. Jeżeli na przykład naszym systemem pomiarowym jest układ wałków o różnych średnicach, to dwa otwory będą w przybliżeniu równe od dołu w naszym systemie pomiarowym, jeśli maksymalne wałki pomiarowe mieszczące się w obu otworach są równe.

Przykłady te dość dobrze oddają intuicje związane z pojęciem zbioru przybliżonego.

Podamy teraz kilka prostych własności zbiorów przybliżonych:

(1) Jeżeli $Y \underset{S}{\simeq} Z$, to $Y \cup Z \underset{S}{\simeq} Y \underset{S}{\simeq} Z$;

(2) Jeżeli $Y \underset{S}{\overline{\simeq}} Z$, to $Y \cap Z \underset{S}{\overline{\simeq}} Y \underset{S}{\overline{\simeq}} Z$;

(3) Jeżeli $Y \underset{S}{\simeq} Y'$ oraz $Z \underset{S}{\simeq} Z'$, to $Y \cup Z \underset{S}{\simeq} Y' \cup Z'$;

(4) Jeżeli $Y \underset{S}{\overline{\simeq}} Y'$ oraz $Z \underset{S}{\overline{\simeq}} Z'$, to $Y \cap Z \underset{S}{\overline{\simeq}} Y' \cap Z'$;

(5) Jeżeli $Y \underset{S}{\overline{\simeq}} Z$, to $-(-Y) \underset{S}{\overline{\simeq}} Z$;

(6) Jeżeli $Y \underset{S}{\simeq} Z$, to $-(-Y) \underset{S}{\simeq} Z$;

(7) Jeżeli $Y \underset{S}{\approx} Z$, to $-(-Y) \underset{S}{\approx} Z$;

(8) Jeżeli $Y \underset{S}{\simeq} Z$, to $Y \cup (-Z) \underset{S}{\simeq} 1$;

(9) Jeżeli $Y \underset{S}{\overline{\simeq}} Z$, to $Y \cap (-Z) \underset{S}{\overline{\simeq}} 0$.

Prawdziwość powyższych własności można wykazać, posługując się podanymi tu własnościami zbiorów przybliżonych. Ich intuicyjny sens nie przedstawia trudności, zostawiamy go więc do przemyślenia Czytelnikowi. Własności te wyjaśniają bliżej, czym są zbiory przybliżone. Można podać ich oczywiście znacznie więcej. Ograniczyliśmy się tu jedynie do niektórych z nich. Możemy teraz ściśle wyrazić wniosek podany na końcu poprzedniego punktu.

Niewiedza eksperta nie zmniejsza dokładności przybliżonej klasyfikacji w systemie informacyjnym S wtedy i tylko wtedy, gdy zbiór niezaklasyfikowanych przez eksperta obiektów jest w przybliżeniu równy w systemie S zbiorowi pustemu ($X_2 \underset{S}{\approx} 0$).

Znaczy to, że niewiedza eksperta jest jakby zamaskowana przez niemożliwość rozróżnienia obiektów za pomocą atrybutów systemu w takim przypadku. Nawet gdyby ekspert potrafił dokładnie rozróżnić wszystkie obiekty i tak za pomocą atrybutów danego systemu informacyjnego nie dałoby się ich wszystkich rozróżnić i zaklasyfikować prawidłowo.

Powyższe rozważania można przenieść również na przypadek niepełnej klasyfikacji wielowartościowej, tzn. na przypadek, w którym ekspert ma za zadanie podzielenie obiektów na n klas ($n > 2$), jednakże w niektórych przypadkach nie potrafi on stwierdzić, do której klasy badany obiekt należy, może natomiast stwierdzić, do których klas obiekt ten nie należy, tj. może on stwierdzić, że obiekt należy do jednej z $k(k < n)$ klas, ale nie wiadomo do której z nich. Problemem tym jednak nie będziemy się tutaj zajmować, gdyż jest on zbliżony do przypadku klasyfikacji dwuwartościowej, który rozważaliśmy do tej pory.

9.4. PRZYPADEK WIELU EKSPERTÓW

W dotychczasowych rozważaniach rozpatrywaliśmy sytuacje, kiedy jeden ekspert oceniał badane obiekty i decydował jak je zaklasyfikować. Istnieją co najmniej dwa powody, dla których taki model nie jest wystarczający.

Pierwszy z nich jest związany z tym, że np. chcielibyśmy przyspieszyć proces klasyfikacji obiektów. Podzielilibyśmy wtedy w dowolny sposób zbiór obiektów między kilku ekspertów, którzy jednocześnie mogliby poklasyfikować swoje obiekty. Powstaje pytanie, czy z przybliżeń klasyfikacji częściowych można zawsze stworzyć przybliżenie klasyfikacji takiej, jaka powstałaby na podstawie klasyfikacji jednego eksperta. Odpowiedź na to pytanie dają własności przybliżeń podane w poprzednim rozdziale, a mianowicie:

(1) $\bar{S}(Y \cup Z) = \bar{S}Y \cup \bar{S}Z$

(2) $\underline{S}(Y \cup Z) \supset \underline{S}Y \cup \underline{S}Z$

Rozważmy ten problem nieco dokładniej. Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym oraz niech $C(X) = \{X_1, X_2, \ldots, X_n\}$ będzie podziałem zbioru X. Przyjmijmy ponadto, że zbiór X został podzielony dowolnie między k ekspertów, tj. podzielony na zbiory $Y_1, Y_2, \ldots, Y_k$. Ekspert numer i dokonuje podziału zbioru Y_i, otrzymując podział $C(Y_i) = \{Y_i^1, Y_i^2, \ldots, Y_i^m\}$, $m \leqslant n$, przy czym $Y_i \in X_j$ oraz $X_j = \bigcup_{i=1}^{k} Y_i^j$. Nasze pytanie możemy obecnie zastąpić pytaniem, czy zachodzi równość:

(3) $\bar{S}X_j = \bigcup_{i=1}^{k} \bar{S}Y_i^j$

(4) $\underline{S}X_j = \bigcup_{i=1}^{k} \underline{S}Y_i^j$

dla każdego $j, i \leqslant j \leqslant n$.

W świetle równości (1) równość (3) jest prawdziwa, natomiast w świetle własności (2) równość (4) nie zachodzi. A więc podział zadania między wielu ekspertów jest możliwy tylko wtedy, gdy chcemy otrzymać na podstawie opisu własności obiektów górne przybliżenie zadanej klasyfikacji. Przy obliczaniu dolnego przybliżenia „zatrudnienie" kilku ekspertów jest więc niedozwolone, nie otrzymamy bowiem zawsze wtedy takich samych rezultatów, jakie otrzymalibyśmy w przypadku użycia tylko jednego eksperta do klasyfikacji.

Drugie zadanie, kiedy jesteśmy zainteresowani zatrudnieniem kilku ekspertów do klasyfikacji, wiąże się z faktem, że chcielibyśmy mieć kilka niezależnych ocen na przykład tego samego materiału medycznego, a następnie stwierdzić, czy oceny ekspertów różnią się istotnie, czy też nie. Główny problem w tak postawionym zadaniu polega na stwierdzeniu, kiedy przybliżenia klasyfikacji dokonanej przez różnych ekspertów mogą być równe. Odpowiedź na to pytanie jest oczywista.

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym oraz niech $C_1(X) = \{X_1^1, X_1^2, \ldots, X_1^n\}$ oraz $C_2(X) = \{X_2^1, X_2^2, \ldots, X_2^n\}$ będą dwoma różnymi podziałami dokonanymi przez dwu różnych ekspertów (dla uproszczenia przyjęliśmy tu przypadek tylko dwu ekspertów).

Powiemy, że podziały $C_1(X)$ i $C_2(X)$ są *w przybliżeniu równe od dołu* w systemie informacyjnym S (symbolicznie $C_1(X) \underset{\underline{S}}{\eqsim} C_2(X)$) wtedy i tylko wtedy, gdy

$$\underline{S}X_1^i = \underline{S}X_2^i, \quad \text{dla każdego } i = 1, 2, \ldots, n.$$

Podobnie powiemy, że podziały $C_1(X)$ oraz $C_2(X)$ są *w przybliżeniu równe od góry* w systemie S (symbolicznie $C_1(X) \underset{\bar{S}}{\simeq} C_2(X)$) wtedy i tylko wtedy, gdy

$$\bar{S}X_1^i = \bar{S}X_2^i, \quad \text{dla każdego } i = 1, 2, \ldots, n.$$

Jeżeli $C_1(X) \underset{\bar{S}}{\simeq} C_2(X)$ oraz $C_1(X) \underset{\underline{S}}{\eqsim} C_2(X)$, to powiemy, że $C_1(X)$ i $C_2(X)$ są w przybliżeniu równe w systemie S i zapiszemy $C_1(X) \underset{S}{\approx} C_2(X)$.

Z podanych definicji oraz z określenia dokładności klasyfikacji podanego w p. 9.2 (wzory (3) i (4)) widać, że różnice w klasyfikacji ekspertów nie odgrywają roli w przybliżonej klasyfikacji wtedy, gdy ich klasyfikacje są w przybliżeniu równe (bądź w przybliżeniu równe od góry lub od dołu) w systemie S.

We wszystkich rozpatrywanych w tym rozdziale zagadnieniach przyjmowaliśmy, że system informacyjny $S = \langle X, A, V, \varrho \rangle$ jest systemem jednowartościowym. Wszystkie te problemy można również rozpatrywać, przyjmując system informacyjny przybliżony (por. rozdz. 6). Otrzymalibyśmy wtedy analizę sytuacji, kiedy również klasyfikacja przybliżona na podstawie własności obiektów nie może być wykonana dokładnie, gdyż wartości niektórych atrybutów obiektów nie mogą być ściśle zidentyfikowane. Nałożenie się tych dwu rodzajów niedokładności, tj. niepełnej klasyfikacji dokonanej przez eksperta oraz niepełnej informacji związanej z nieokreślonością wartości niektórych atrybutów — prowadzi do interesujących wniosków, jednakże, z uwagi na konieczność wprowadzenia wielu nowych pojęć, problematyką tą nie będziemy się zajmowali w niniejszej książce.

10. O PROCESIE UCZENIA SIĘ

10.1. UWAGI WSTĘPNE

Na problematykę poruszaną w ostatnich dwu rozdziałach można również spojrzeć z nieco innego punktu widzenia, a mianowicie z punktu widzenia procesu uczenia się. Jeżeli klasyfikację zbioru dokonaną przez eksperta traktować jako wyraz jego wiedzy, a dokonanie przybliżenia powyższej klasyfikacji na podstawie własności przedmiotów, jako przyswojenie wiedzy eksperta, to możemy wtedy mówić, że osoba, która dokonała klasyfikacji przybliżonej „posiadła" częściowo wiedzę eksperta. Na przykład jeżeli wiedza lekarza specjalisty pozwala mu na prawidłowe stawianie diagnozy medycznej i chce on nauczyć studentów medycyny stawiania diagnozy, niewątpliwie jedną z możliwości będzie tu próba scharakteryzowania choroby przez jej objawy (chociaż na pewno nie jest to wystarczające do właściwej diagnozy). Posługując się opisem choroby w postaci opisów występujących w niej objawów, student będzie mógł uzyskać w stawianiu diagnozy rezultaty zbliżone do doświadczonego lekarza — chociaż, jak wiemy z poprzednich rozważań, pełnej zgodności z diagnozą specjalisty w ogólnym przypadku tą metodą osiągnąć się nie da. Jest to więc niewątpliwie pewna metoda przekazywania wiedzy (chociaż nie jedyna) — a więc uczenia się lub też nauczania.

Formułując to nieco inaczej: nauczyciel klasyfikuje obiekty na podstawie swej wiedzy, natomiast student stara się znaleźć cechy charakterystyczne każdej klasy obiektów podanej przez nauczyciela, tak aby następnie za pomocą tych własności przeprowadzić podobną klasyfikację jak nauczyciel.

10.2. PROCES UCZENIA

Przeanalizujemy obecnie ten problem nieco dokładniej.

Niech $S = \langle X, A, V, \varrho \rangle$ będzie systemem informacyjnym oraz niech $C(X)$ będzie podziałem zbioru X dokonanym przez eksperta (nauczyciela) oraz niech $\overline{C}_S(X)$ oraz $\underline{C}_S(X)$ będą przybliżeniami podziału $C(X)$ w systemie S, dokonanymi przez osobę uczącą się. W ten sposób możemy przyjąć, że podział $C(X)$ reprezentuje wiedzę nauczyciela, podziały zaś $\overline{C}_S(X)$ oraz $\underline{C}_S(X)$ wiedzę studenta.

Załóżmy, że zadanie studenta polega teraz na tym, że otrzymuje on nowe obiekty (nie należące do zbioru X) i ma je na podstawie atrybutów systemu S, zaliczyć do właściwych klas. Decyzje studenta są kontrolowane przez nauczyciela, który dokonuje niezależnej klasyfikacji.

Dodając więc nowy obiekt do zbioru X, tworzymy nowy system informacyjny S', nowy podział $C_{S'}(X')$ oraz nowe jego przybliżenia $\overline{C}_{S'}(X')$ i $\underline{C}_{S'}(X')$.

Tak rozumiany proces uczenia się można więc opisać za pomocą ciągu systemów informacyjnych

$$S^0 = \langle X^0, A, V, \varrho^0 \rangle$$

$$S^1 = \langle X^1, A, V, \varrho^1 \rangle$$

$$\ldots$$

$$S^n = \langle X^n, A, V, \varrho^n \rangle$$

ciągu podziałów

$$C_{S^0}(X^0) = \{X_1^0, X_2^0, \ldots, X_{k_0}^0\}$$

$$C_{S^1}(X^1) = \{X_1^1, X_2^1, \ldots, X_{k_1}^1\}$$

$$\ldots$$

$$C_{S^n}(X^n) = \{X_1^n, X_2^n, \ldots, X_{k_n}^n\}$$

ciągu przybliżeń dolnych

$$\underline{C}_{S^0}(X^0) = \{\underline{S}X_1^0, \underline{S}X_2^0, \ldots, \underline{S}X_{k_0}^0\}$$

$$\underline{C}_{S^1}(X^1) = \{\underline{S}X_1^1, \underline{S}X_2^1, \ldots, \underline{S}X_{k_1}^1\}$$

$$\ldots$$

$$\underline{C}_{S^n}(X^n) = \{\underline{S}X_1^n, \underline{S}X_2^n, \ldots, \underline{S}X_{k_n}^n\}$$

oraz ciągu przybliżeń górnych

$$\overline{C}_{S^0}(X^0) = \{\bar{S}X_1^0, \bar{S}X_2^0, \ldots, \bar{S}X_{k_0}^0\}$$

$$\overline{C}_{S^1}(X^1) = \{\bar{S}X_1^1, \bar{S}X_2^1, \ldots, \bar{S}X_{k_1}^1\}$$

$$\ldots$$

$$\overline{C}_{S^n}(X^n) = \{\bar{S}X_1^n, \bar{S}X_2^n, \ldots, \bar{S}X_{k_n}^n\}$$

przy czym $S^0 = S$, $X^0 = X$ itd.

Zauważmy, że każdy system S_i jest podsystemem z ograniczonymi obiektami systemu S_j, gdy $i < j$. Nasze zadanie możemy więc również sformułować nieco inaczej.

Dany jest system informacyjny $S = \langle X, A, V, \varrho \rangle$ oraz podział $C(X)$. Proces uczenia polega na wybraniu pewnego podsystemu początkowego S^0 systemu S, a następnie utworzeniu ciągu systemów S^0, S^1, ..., S^n, takiego że S^i jest podsystemem systemu S^{i+1} oraz $S^n = S$, oraz konstruowaniu odpowiadającym podsystemem podziałów ewentualnie ich przybliżeń, tj. zaliczenia każdego nowego obiektu do właściwej mu klasy bloku podziału, tak by w ostatnim kroku uczenia uzyskać podział jak najbardziej zbliżony do podziału eksperta.

10.3. DOKŁADNOŚĆ UCZENIA SIĘ — OSZACOWANIE OD DOŁU

Głównym problemem, który będziemy tu rozważać, jest problem: jak zależy dokładność przybliżonego podziału od „długości" nauczania lub mówiąc niezbyt ściśle, jak zależy „jakość" nauczania od „długości" nauczania.

Jakość nauczania można mierzyć wprowadzonymi w poprzednim rozdziale współczynnikami dokładności podziału oraz dyspersji podziału. Będziemy interesowali się, jak zmieniają się te współczynniki w zależności od długości nauczania, tj. jak zależą one od numeru i systemu $S^i(0 \leqslant i \leqslant n)$. Rozpatrzymy tylko współczynnik dokładności, gdyż współczynnik dyspersji w prosty sposób zależy od współczynnika dokładności. Założymy przy tym, że X^{i+1} różni się od X^i jednym elementem. Ponieważ

$$(1)\quad \alpha_{S^{i+1}} C(X^{i+1}) = \frac{\sum_{j=1}^{k_{i+1}} \operatorname{card}(\underline{S}^{i+1}(X_j^{i+1}))}{\operatorname{card}(X^{i+1})}$$

gdzie

$$C_{S^{i+1}}(X^{i+1}) = \{X_1^{i+1}, X_2^{i+1}, \ldots, X_{k_{i+1}}^{i+1}\}$$

oraz

$$(2)\quad \alpha_{S^i} C(X^i) = \frac{\sum_{j=1}^{k_i} \operatorname{card}(\underline{S}^i(X_j^i))}{\operatorname{card}(X^i)}$$

gdzie

$$C_{S^i}(X^i) = \{X_1^i, X_2^i, \ldots, X_{k_i}^i\}$$

więc dla zrealizowania naszego zadania musimy współczynnik (1) wyrazić jako funkcję współczynnika (2).

Zauważmy, że zgodnie z naszym założeniem w każdym kroku procesu uczenia przybywa jeden nowy obiekt y, tj.

$$\operatorname{card}(X^{i+1}) = \operatorname{card}(X^i) + 1 \text{ dla każdego } i$$

oraz że nowy obiekt y jest „dodawany" do zbioru elementarnego $X^i_{\varrho_y}$ systemu S^i, tworząc w ten sposób nowy zbiór elementarny $X^{i+1}_{\varrho_y} = X^i_{\varrho_y} \cup \{y\}$, systemu S^{i+1}. W szczególności $X^i_{\varrho_y}$ może być zbiorem pustym. Przypominamy, że

$$X^i_{\varrho_y} = \{x \in X^i : \varrho_x = \varrho_y\}$$

Dla rozwiązania rozpatrywanego zadania musimy rozpatrzeć cztery następujące przypadki:

(a) Zbiór elementarny $X^i_{\varrho_y}$ jest zbiorem wewnętrznym podziałów $C_{S^i}(X^i)$ oraz $C_{S^{i+1}}(X^{i+1})$. Oznacza to, że obiekt y może być właściwie zaklasyfikowany na podstawie dotychczasowego doświadczenia.

(b) Zbiór elementarny $X^i_{\varrho_y}$ jest zbiorem brzegowym podziału $C_{S^i}(X^i)$, a w konsekwencji również zbiorem brzegowym podziału $C_{S^{i+1}}(X^{i+1})$. Oznacza to, że obiekt y nie może zostać właściwie zaklasyfikowany w podziale $C_{S^{i+1}}(X^{i+1})$ na podstawie dotychczasowego doświadczenia.

c) Zbiór elementarny $X^i_{\varrho_y}$ jest pusty, tzn. że w dotychczasowym doświadczeniu obiekty o własnościach takich jak obiekt y.

(d) Zbiór elementarny $X^i_{\varrho_y}$ jest zbiorem wewnętrznym w podziale $C_{S^i}(X^i)$, a w podziale $C_{S^{i+1}}(X^{i+1})$ zbiorem brzegowym. Obiekt y możemy traktować więc jako kontrprzykład — nie potwierdzający dotychczasowego doświadczenia. Wyjaśnimy to dokładniej przy omawianiu tego punktu.

W przypadku (a) sprawa jest bardzo prosta:

$$(3)\quad \alpha_{S^{i+1}}C(X^{i+1}) = \frac{\sum_{j=1}^{k_i} \operatorname{card}(\underline{S}^{i+1}(X^i_j))+1}{\operatorname{card}(X^i)+1}$$

gdyż w tym przypadku wzrośnie o jeden liczba wszystkich obiektów w systemie S^{i+1} oraz o jeden wzrośnie liczba właściwie zaklasyfikowanych obiektów. Gdyby więc w całym procesie uczenia wszystkie obiekty były właściwie zaklasyfikowane, dokładność uczenia rosłaby jednostajnie z każdym nowym obiektem według zależności (3).

W przypadku (b) sprawa jest również bardzo prosta:

$$(4)\quad \alpha_{S^{i+1}}C(X^{i+1}) = \frac{\sum_{j=1}^{k_i} \operatorname{card}(\underline{S}^i(X^i_j))}{\operatorname{card}(X^i)+1}$$

gdyż nowego obiektu y nie potrafimy prawidłowo zaklasyfikować. A więc dokładność w stosunku do poprzedniego kroku zmalała.

W przypadku (c) sytuacja jest dokładnie taka sama, jak w przypadku (a) i można tu zastosować wzór (3).

Najciekawszy jest przypadek (d). Wyobraźmy sobie następującą sytuację: lekarz specjalista zakwalifikował pacjenta jako zdrowego, student zaś na podstawie objawów i dotychczasowego procesu uczenia zakwalifikował tegoż pacjenta jako chorego. Oznacza to, że dotychczasowy opis choroby był niedostateczny i do uzyskania prawidłowego opisu należy włączyć nowy przypadek i na podstawie nowego zbioru przypadków określić ponownie objawy rozpatrywanej choroby. W konsekwencji oznacza to, że zbiór wszystkich pacjentów zdrowych, mających taki sam opis, jak nowy pacjent uznany za chorego, musi być wyłączony z dolnego przybliżenia pacjentów zdrowych i znaleźć się w zbiorze nowego podziału.

Ogólnie możemy to sformułować w następujący sposób: niech y będzie obiektem różnie zaklasyfikowanym przez ucznia i nauczyciela. Wtedy zbiór $X^i_{\varrho_y}$ jest elementarnym zbiorem wewnętrznym podziału $C_{S^i}(X^i)$, zaś $X^{i+1}_{\varrho_y} = X^i_{\varrho_y} \cup \{y\}$ jest elementarnym zbiorem brzegowym podziału $C_{S^{i+1}}(X^{i+1})$. W rezultacie powoduje to, że zamiast wzoru (1) mamy wzór następującej postaci:

$$(5)\quad \alpha_{S^{i+1}}C(X^{i+1}) = \frac{\sum_{j=1}^{k_i} \operatorname{card}(\underline{S}^i(X_j^i)) - \operatorname{card}(X_{\varrho_y}^i)}{\operatorname{card}(X^i)+1}$$

Element y o własnościach takich, jak podano powyżej, będziemy nazywać kontrprzykładem. Ze wzoru (5) widać, że kontrprzykład bardzo pogarsza efektywność nauczania. Wielkość tego pogorszenia zależy od wielkości zbioru $X_{\varrho_y}^i$, tj. od tego, jak dużo mieliśmy obiektów w procesie nauczania o własnościach takich jak obiekt y, lecz należących do innego bloku podziału niż obiekt y.

10.4. UWAGI KOŃCOWE

Podane w tym rozdziale rozważania nie mają na celu analizy procesu uczenia się ani też nauczania u człowieka, a jedynie stanowią ilustrację możliwości zastosowań wprowadzonych poprzednio pojęć do rozwiązania niektórych problemów tzw. sztucznej inteligencji. Mogą one być więc przydatne w budowie i analizie algorytmów realizowanych w tzw. programach i maszynach samouczących się. Algorytmy takie mają zastosowanie w rozpoznawaniu obrazów, znaków, mowy, gry w szachy itd. Oczywiście przeprowadzona tu analiza jest dalece niewystarczająca do tego celu, daje ona jedynie informacje, że pojęcie systemu informacyjnego i jego języka mogą znaleźć również zastosowania w rozwiązywaniu tego rodzaju problemów.

Zwróćmy tu również uwagę na związek omawianych problemów z tzw. logiką indukcji, której główny problem polega na analizie mechanizmów stawiania hipotez na podstawie przykładów. Na temat ten istnieje bogata literatura (patrz np. prace [13], [14], [15]).

Reprezentowane w tej książce podejście jest bliżej omawiane w pracy [35].

ZAKOŃCZENIE

W książce sprecyzowano niektóre podstawowe pojęcia dotyczące systemów informacyjnych oraz rozważań indukcyjnych w oparciu o pojęcia systemu informacyjnego oraz języka systemu informacyjnego. Wskazano również możliwości zastosowań praktycznych uzyskanych rezultatów.

W istocie jednak wiele z poruszanych tu zagadnień nie zbadano dość szczegółowo, np. pojęcie systemów informacyjnych wielowartościowych, przybliżonych, stochastycznych. W rzeczywistości z praktycznego punktu widzenia ciekawsze są nie tyle różnego rodzaju rozważane tu systemy informacyjne, lecz raczej takie systemy, w których mogą występować jednocześnie różnego rodzaju atrybuty, np. jednowartościowe, wielowartościowe, przybliżone, stochastyczne. Punktem wyjścia do badań tego rodzaju systemów z „mieszanymi atrybutami", mogłoby być pojęcie systemu, zaproponowane w p. 7.4, gdzie system informacyjny rozumie się jako parę: zbiór obiektów oraz pewna relacja równoważności. Relacja ta byłaby dokładniej zdefiniowana przez atrybuty, niezależnie od tego, jakiego są one rodzaju. Kłopotem istotnym w tego rodzaju podejściu mogłaby być strona logiczna zagadnienia, musielibyśmy bowiem w takim przypadku mieć zbiór aksjomatów pozwalający „operować" językiem, w którym jednocześnie występują atrybuty różnego rodzaju, co nie wydaje się sprawą prostą. Jednakże z praktycznego punktu widzenia trudność taką można by pominąć, gdyż nie wydaje się, aby dla realizacji systemów konieczne było wchodzenie zbyt głęboko w logiczną strukturę języka z mieszanymi atrybutami.

Jeżeli chodzi o problematykę poruszaną w rozdz. 8÷10, to nierozwiązanych problemów jest znacznie więcej. Byłoby przede wszystkim interesujące porównanie wynikające z naszych rozważań metod klasyfikacji z istniejącymi metodami, których jest bardzo wiele i mają one bogatą literaturę oraz zastosowania. Nasuwa się również pytanie, jaki jest stosunek zaproponowanego tu modelu uczenia się do innych znanych modeli. Problematyka ta ma również bardzo bogatą tradycję i literaturę. Byłoby rzeczą interesującą rozwinięcie myśli podanej w rozdz. 9, dotyczącej tego, jak wpływa na dokładność uczenia się „niewiedza" eksperta, oraz „niewiedza" ucznia, tj. zbadanie zastosowania przybliżonych i stochastycznych systemów informacyjnych do tej problematyki. Wydaje się również, że w świetle wprowadzonych pojęć byłoby interesujące spojrzenie z proponowanego tu punktu widzenia na logikę indukcji. Zastosowanie proponowanego podejścia może nie miałoby większego znaczenia dla badań matematycznych nad logiką indukcji, jednakże na pewno byłoby ono użyteczne z punktu widzenia automatyzacji rozumowań indukcyjnych.

Wreszcie ostatni problem, to problem zbiorów przybliżonych. Pojęcie to, które jest naturalną konsekwencją użytego pojęcia systemu informacyjnego, wydaje się być bardzo interesujące zarówno z teoretycznego punktu widzenia, jak i praktycznych zastosowań, nie tylko w dziedzinach omawianych w tej książce. Pojęcie zbioru przybliżonego jest nieco zbliżone do pojęcia zbioru rozmytego i byłoby rzeczą interesującą zbadanie związków między tymi dwoma pojęciami zbioru. Z drugiej strony pojęcie zbioru przybliżonego wiąże się bardzo z pojęciem tolerancji i wydaje się, że może być ono substytutem tego pojęcia. Byłoby rzeczą interesującą podjęcie próby sformułowania teorii tolerancji w terminach zbiorów przybliżonych. Wydaje się, że takie podejście jest bardziej naturalne do formułowań i analizy problemów występujących w tej teorii. Wreszcie z proponowanym pojęciem zbioru przybliżonego wiąże się problematyka logiki modalnej. Gruntowane zbadanie tych związków dałoby na pewno lepszy wgląd w istotę zbiorów przybliżonych, a w konsekwencji ich zastosowań.

Tak więc w książce tej chyba więcej problemów postawiono niż rozwiązano. Mam nadzieję, że nie wszyscy Czytelnicy uznają to za wadę.

Dodatek A.
POJĘCIA MATEMATYCZNE UŻYWANE W PRACY

A1. NOTACJA LOGICZNA

Jeżeli p, q są zdaniami, to $p+q, p\cdot q, \sim p$ oznaczają odpowiednio alternatywę, koniunkcję oraz negację zdań p, q. Ponadto $p \rightarrow q$ oznacza implikację zdefiniowaną jako $\sim p+q$, oraz $p \leftrightarrow q$ oznacza równoważność, zdefiniowaną jako $(p\rightarrow q)\cdot(q\rightarrow p)$. Symbole $\bigwedge$, $\bigvee$ oznaczają odpowiednio duży i mały kwalifikator, tj. zwroty „dla każdego" oraz „istnieje".

A2. ZBIORY

Jeżeli obiekt x *należy* do zbioru X, to zapiszemy $x \in X$; jeżeli obiekt x *nie należy* do zbioru X, zapiszemy $\sim x \in X$ lub $x \notin X$.

Zbiór pusty oznaczamy przez $\emptyset$. Zbiór, do którego należą elementy $x_1, x_2, \ldots, x_n$, oznaczamy $\{x_1, x_2, \ldots, x_n\}$. Jeżeli X jest zbiorem wszystkich elementów spełniających warunek W, piszemy $X = \{x:, W(x)\}$.

Mocą zbioru X nazywamy liczbę jego elementów i oznaczamy $\mathrm{card}(X)$.

Sumą zbiorów X i Y nazywamy zbiór, do którego należą wszystkie elementy zbioru X oraz wszystkie elementy zbioru Y. Sumę zbiorów X i Y oznaczamy przez $X \cup Y$.

Iloczynem zbiorów X i Y nazywamy zbiór, którego elementami są elementy należące zarówno do zbioru X, jak i do zbioru Y. Iloczyn zbiorów oznaczamy przez $X \cap Y$.

Różnicą zbiorów X i Y nazywamy zbiór elementów należących do zbioru X i nie należących do zbioru Y. Różnicę zbiorów X i Y oznaczamy przez $X - Y$.

Jeżeli U jest ustalonym zbiorem, to $U - X$ nazywamy *dopełnieniem zbioru* X *do zbioru* U i oznaczamy przez $-X$.

Mówimy, że zbiór X jest *podzbiorem zbioru* Y (lub X jest *zawarty* w Y, lub Y *zawiera* X), gdy każdy element zbioru X należy do zbioru Y. Jeżeli X jest podzbiorem zbioru Y, zapisujemy $X \subset Y$. Symbol $\subset$ nazywamy znakiem inkluzji.

Rodzinę wszystkich podzbiorów zbioru X oznaczamy przez $\mathscr{P}(X)$.

Jeżeli X nie jest podzbiorem Y, zapisujemy $X \not\subset Y$ lub $\sim X \subset Y$.

Zbiory X, Y są równe, gdy mają te same elementy. Jeżeli X i Y są równe, zapisujemy $X = Y$.

A3. RELACJE

Produktem (*iloczynem*) *kartezjańskim* zbiorów X i Y nazywamy zbiór wszystkich par uporządkowanych (x, y), takich że $x \in X$ i $y \in Y$. Produkt kartezjański X i Y oznaczamy przez $X \times Y$.

Jeżeli $X = Y$, to zamiast $X \times Y$ piszemy też X^2.

Relacją dwuczłonową w produkcie kartezjańskim $X \times Y$ nazywamy dowolny podzbiór produktu $X \times Y$.

Relacje oznaczamy małymi literami greckimi.

Jeżeli ϱ jest relacją dwuczłonową w produkcie X^2, to mówimy, że ϱ jest *relacją dwuczłonową w* X.

Jeżeli ϱ jest relacją dwuczłonową w $X \times Y$, to zamist $(x, y) \in \varrho$ będziemy pisać również $\varrho(x, y)$.

Dziedziną dwuczłonowej relacji ϱ w $X \times Y$ nazwiemy zbiór tych wszystkich elementów $x \in X$, dla których istnieje $y \in Y$ takie, że $\varrho(x, y)$. Dziedzinę relacji ϱ oznaczamy przez $D\varrho$. Możemy więc zapisać

$$D\varrho = \{x \in X : \bigvee_{y \in Y} \varrho(x, y)\}$$

Przeciwdziedziną relacji ϱ w $X \times Y$ nazwiemy zbiór tych wszystkich elementów $y \in Y$, dla których istnieje takie $x \in X$, że $\varrho(x, y)$. Przeciwdziedzinę relacji oznaczamy przez $R\varrho$. Możemy więc zapisać:

$$R\varrho = \{y \in Y : \bigvee_{x \in X} \varrho(x, y)\}$$

Relację dwuczłonową $\varrho \subset X \times Y$ nazywamy *zwrotną*, gdy dla każdego $x \in X$, $\varrho(x, x)$.

Relację dwuczłonową $\varrho \subset X \times Y$ nazywamy *przeciwzwrotną*, jeżeli dla każdego $x \in X$, $\sim\varrho(x, x)$.

Relację dwuczłonową $\varrho \subset X^2$ nazywamy *symetryczną*, gdy dla każdego $x \in X$ jeżeli $\varrho(x, y)$, to również $\varrho(y, x)$, symbolicznie:

$$\bigwedge_{x,y \in X} \varrho(x, y) \to \varrho(y, x)$$

Relację $\varrho \subset X^2$ nazywamy *przeciwsymetryczną* w X, gdy dla każdego $x, y \in X$ jeżeli $\varrho(x, y)$ to również $\sim\varrho(y, x)$, symbolicznie.

Relację $\varrho \subset X^2$ nazywamy *antysymetryczną*, jeśli warunki $\varrho(x, y)$ oraz $\varrho(y, x)$ pociągają $x = y$, symbolicznie

$$\bigwedge_{x,y,\in X} \varrho(x, y) \wedge \varrho(x, y) \to x = y$$

Relację $\varrho \subset X^2$ nazywamy *przechodnią*, jeżeli dla każdego $x, y, z \in X \varrho(x, y)$ oraz $\varrho(y, z)$ pociąga $\varrho(x, z)$, symbolicznie:

$$\bigwedge_{x,y,z \in X} \varrho(x, y) \wedge \varrho(y, z) \to \varrho(x, z)$$

A4. RELACJE RÓWNOWAŻNOŚCI

Relacje $\varrho \subset X^2$ nazywamy *relacją równoważności*, gdy jest ona zwrotna, symetryczna i przechodnia, tj.

(1) $\varrho(x, x)$ dla każdego $x \in X$;

(2) $\varrho(x, y) \to \varrho(y, x)$ dla każdego $x, y \in X$;

(3) $\varrho(x, y) \wedge \varrho(y, z) \to \varrho(x, z)$ dla każdego $x, y, z \in X$.

Niech $\varrho \in X^2$ będzie relacją równoważności w X i niech $B(x)$ oznacza zbiór wszystkich elementów $y \in X$ będących w relacji z x, tj.

$$B(x) = \{y \in X: \varrho(x, y)\}$$

Zbiory $B(x)$, dla $x \in X$, nazywamy *klasami równoważności* (albo *abstrakcji*) relacji ϱ.

Dowolna relacja równoważności $\varrho \subset X^2$ określa podział zbioru X na niepuste rozłączne podzbiory, tj. na klasy równoważności relacji $\varrho \subset X^2$, tj.

(4) $B(x) \cap B(y) = \emptyset$, gdy $\sim\varrho(x, y)$

(5) $\bigcup_{x \in X} B(x) = X$

A5. RELACJE PORZĄDKU

Relację $\varrho \subset X^2$ nazywamy *relacją porządkującą* zbiór X, jeśli jest zwrotna, przechodnia i antysymetryczna, tj.

(1) $\varrho(x, x)$ dla każdego $x \in X$;

(2) $\varrho(x, y) \wedge \varrho(y, z) \to \varrho(x, z)$ dla każdego $x, y, z \in X$.

Jeżeli $\varrho \subset X^2$ jest relacją porządkującą zbiór X, to powiemy, że *ϱ porządkuje X*, parę (X, ϱ) zaś nazwiemy *zbiorem uporządkowanym*.

Niech X będzie zbiorem uporządkowanym. Powiemy, że x poprzedza $y (x, y \in X)$, gdy $\varrho(x, y)$ oraz $x \neq y$.

Element $x \in X$ nazwiemy *maksymalnym*, jeżeli nie poprzedza on żadnego elementu w zbiorze X.

Element $x \in X$ nazwiemy *elementem minimalnym*, gdy nie poprzedza go żaden element w zbiorze X.

A6. FUNKCJE

Niech $\varrho \subset X \times Y$. Relacje ϱ nazwiemy funkcją, gdy dla każdego $x \in X$ istnieje dokładnie jeden element $y \in Y$, taki że $\varrho(x, y)$.

Funkcję będziemy oznaczać małymi literami łacińskimi. Jeżeli $f \subset X \times Y$, to będziemy pisać

$$f: X \to Y$$

i powiemy, że *f odwzorowuje* (*przekształca*) zbiór X w Y.

Jeżeli $f\colon X \to Y$, to zamiast $f(x, y)$ będziemy pisać $f(x) = y$ i $f(x)$ będziemy nazywać *wartością* funkcji f dla argumentu x, zamiast $f(x)$ będziemy też pisać f_x.

Zbiór tych elementów $x \in X$, dla których funkcja f jest określona, nazwiemy *dziedziną funkcji* f i oznaczymy przez D_f.

Zbiór wszystkich elementów $y \in Y$, które są wartościami funkcji f, tj. tych elementów $y \in Y$, dla których istnieje takie $x \in X$, że $f(x) = y$, nazwiemy *przeciwdziedziną* funkcji f i oznaczymy przez R_f.

Dodatek B.
UWAGI O REALIZACJI SYSTEMÓW INFORMACYJNYCH METODĄ SKŁADOWYCH ATOMOWYCH

B1. UWAGI WSTĘPNE

Z podanego w rozdz. 2 twierdzenia o postaci normalnej wynika nowa możliwość realizacji systemów wyszukiwania informacji. Jeżeli dane pogrupujemy w pamięci w grupy odpowiadające zbiorom elementarnym, to odpowiedź na dowolne pytanie można znaleźć, sprowadzając najpierw zadane pytanie do postaci normalnej i następnie wyszukać zbiory elementarne odpowiadające wszystkim termom prostym występującym w postaci normalnej pytania.

Taki algorytm wyszukiwania ma wiele zalet w stosunku do algorytmów używanych w dotychczasowych systemach wyszukiwania. Najważniejsze z nich to możliwość zoptymalizowania dostępów do pamięci masowej (dyskowej lub taśmowej) oraz możliwość uniknięcia wykonywania mało efektywnych operacji na zbiorach danych, takich jak iloczyn, uzupełnienie i sortowanie.

Celowość stosowania proponowanej metody zależy od wielu czynników, przede wszystkim od wielkości bazy danych, rodzaju pytań, stosunku szybkości pamięci wolnej do pamięci szybkiej.

Ogólnie, im większe są zbiory danych, im bardziej złożone pytania oraz im większy jest stosunek szybkości (wybierania i odczytu) pamięci wolnej do szybkiej (operacyjnej), tym bardziej celowe jest stosowanie wyszukiwania informacji metodą składowych atomowych.

Inną jeszcze zaletą omawianej metody wydaje się być możliwość współbieżnego (równoległego) wykonywania operacji wyszukiwania, szczególnie w systemach wielodostępnych. Równoległość ta może być realizowana programowo lub sprzętowo (np. przez niezależne jednostki wyszukujące).

Stosowanie metody składowych atomowych daje możliwość uzyskania odpowiedzi na niektóre rodzaje pytań bez konieczności sięgania do zbiorów danych, a jedynie na podstawie formalnych przekształceń pytań. Jest to konsekwencja ścisłego sformułowania języka zapytań oraz jego składni i semantyki.

B2. ZBIORY ELEMENTARNE

Podstawowym problemem w metodzie składowych atomowych jest organizacja danych w postaci zbiorów elementarnych. Od tej organizacji zależy celowość stosowania metody oraz w dużym stopniu jej efektywność. Podstawowym parametrem jest tutaj liczba zbiorów elementarnych oraz ich wielkość. Dokładne oszacowanie tych parametrów nie wydaje się możliwe drogą analityczną, lecz jedynie metodami symulacyjnymi.

Istnieje tu pewna sprzeczność między wymaganiami technicznymi a wymaganiami logicznymi. Wymagania techniczne prowadzą raczej do niezbyt dużej liczby zbiorów elementarnych o tak dobranych wielkościach, aby czasy dostępu do zbiorów elementarnych stanowiących odpowiedzi na przeciętne pytania i ich odczytu były możliwie małe. Natomiast z własności atrybutów i definiowanej przez nie relacji równoważności wynika bardzo duża liczba małych zbiorów elementarnych, co od razu w widoczny sposób prowadzi do bardzo małej, praktycznie nie do przyjęcia, efektywności algorytmów wyszukiwania.

Dla uniknięcia tej sprzeczności zastosowano w dotychczasowych realizacjach tej metody wyszukiwanie dwustopniowe. Ideę takiego wyszukiwania podano w rozdz. 3. Polega ona na zastosowaniu nie wszystkich, lecz tylko niektórych odpowiednio wybranych atrybutów do utworzenia zbiorów elementarnych, tak aby miały one odpowiednie parametry techniczne. Zbiór uzyskany na podstawie tak wybranych atrybutów nie będzie oczywiście odpowiedzią właściwą. Dla uzyskania właściwej odpowiedzi należy dokonać wyboru obiektów według dalszych atrybutów w zbiorze otrzymanym w pierwszym etapie wyszukiwania.

Tak więc w istocie w pierwszym etapie wyszukiwania stosuje się metodę składowych atomowych, w drugim zaś można zastosować dowolną metodę, np. metodę przeglądu zupełnego. Pierwszy krok pozwala więc na zredukowanie obszaru pamięci, w którym następuje właściwe wyszukiwanie odpowiedzi. Współczynnik tej redukcji może być rzędu kilku do kilkuset. O jego wielkości decydują charakterystyki pamięci maszyny oraz rodzaj pytań „średnio" stawianych do systemu.

Najwłaściwszym z istniejących aktualnie rodzajem pamięci do realizacji metody składowych atomowych, wydaje się pamięć dyskowa. Zbiory elementarne powinny stanowić wtedy całe ścieżki lub cylindry bądź ich wielokrotności.

Pamięć taśmowa w zasadzie nie wydaje się przydatna do tego celu z uwagi na niemożliwość szybkiego dostępu do wybranej porcji taśmy, stanowiącej zbiór elementarny. Stosowanie jej wydaje się jedynie celowe w przypadku, gdy cała taśma może stanowić zbiór elementarny i dysponujemy jednoczesnym dostępem do kilku taśm. Wtedy wyszukiwanie dwuetapowe polega na wyznaczeniu w pierwszym kroku taśm, w których należy szukać odpowiedzi, a następnie w drugim kroku metodą przeglądu zupełnego równoległe wyszukanie we wstępnie wyznaczonych taśmach ostatecznej odpowiedzi.

Zastosowanie takiej metody wydaje się być uzasadnione w przypadku bardzo dużych baz danych. W przypadku bardzo dużych baz danych również całe dyski mogłyby być traktowane jako zbiory elementarne, tj. w pierwszym etapie wyszukiwania zostałyby wyznaczone dyski, na których znajduje się odpowiedź, a następnie szczegółowa odpowiedź byłaby już wyszukiwana w wyznaczonych dyskach równolegle. Tutaj w przeciwieństwie do pamięci taśmowych również drugi etap wyszukiwania może być realizowany metodą składowych atomowych.

Reasumując, podział danych na zbiory elementarne jest podstawowym problemem w proponowanej metodzie i jest on głównie uzależniony od parametrów technicznych przez charakter pytań stawianych najczęściej do systemu.

B3. WYSZUKIWANIE ZBIORÓW ELEMENTARNYCH

Dla wyszukania zbiorów elementarnych wchodzących w skład odpowiedzi na pytanie zadane w postaci normalnej, musimy dysponować odpowiednimi algorytmami pozwalającymi na podstawie termu prostego znaleźć odpowiadający mu zbiór elementarny.

Najprościej zadanie to można zrealizować, tworząc tablicę adresową, w której są podane adresy początków zbiorów elementarnych oraz odpowiadające im termy proste. Tak więc w celu znalezienia zbioru elementarnego odpowiadającego danemu termowi prostemu należy w tablicy znaleźć żądany term prosty, a następnie odpowiadający mu adres zbioru elementarnego.

Rozwiązanie takie na ogół nie jest efektywne, wymaga bowiem przeszukiwania tablicy adresowej metodą przeglądu zupełnego celem znalezienia w niej termu prostego, występującego w pytaniu. Istnieje ulepszona wersja tej metody, która pozwala na efektywniejsze znalezienie adresów zbiorów elementarnych niż w omawianym przypadku.

Inna, bardziej efektywna metoda, polega na odpowiedniej numeracji termów prostych i posługiwaniu się tablicą adresową, w której adresom zbiorów elementarnych odpowiadają nie termy proste, lecz ich numery. Zasadniczą sprawą jest tutaj efektywna numeracja termów prostych. Przykład takiej numeracji podano w pracy [63]. Została ona zastosowana w realizacji systemu wyszukiwania przez Margańskiego [67].

Można również wyszukać zbiory elementarne wchodzące w skład odpowiedzi na zadane pytanie bez sprowadzania go do postaci normalnej. Jest wtedy konieczne utworzenie specjalnej tablicy zawierającej informacje, jakie zbiory elementarne wchodzą w skład każdego deskryptora. Tablica taka może oczywiście zawierać od razu również adresy tych zbiorów. Wtedy wszystkie działania teoriomnogościowe występujące w pytaniu można wykonać na zbiorach elementarnych (ściślej ich adresach) występujących w tablicy, a nie na zbiorach danych, co również pozwala na efektywniejsze realizowanie procesu wyszukiwania.

Efektywność takiego algorytmu w dużym stopniu zależy od charakteru pytań i celowość jego użycia może być oceniana na podstawie dokładnej analizy jego efektywności.

B4. POSTAĆ NORMALNA

Sprowadzanie pytań do postaci normalnej w metodzie składowych atomowych ma zasadnicze znaczenie dla tej metody. Od efektywności tego algorytmu zależy w istotny sposób efektywność całej metody. Jeżeli bowiem algorytm ten jest zbyt powolny, czas uzyskany na zmniejszenie dostępów do pamięci wolnej straci się na sprowadzenie pytania do postaci normalnej i stosowanie tej metody nie może być wtedy celowe.

Istnieje wiele algorytmów sprowadzania pytań tego rodzaju do postaci normalnej, jednak do tej pory nie został dostatecznie przebadany problem ich efektywności. Wszystkie one w zasadzie bazują na stosowaniu aksjomatów algebry Boole'a oraz aksjomatów specyficznych, w standardowy sposób. Przy maszynach o dostatecznie dużej szybkości operacyjnej problem ten ma mniejsze znaczenie niż przy maszynach wolnych, gdzie sprawa efektywności sprowadzania pytań do postaci normalnej jest zasadniczej wagi.

B5. WIELODOSTĘP

Wydaje się, że metoda składowych atomowych powinna być przydatna w wielodostępnych systemach wyszukiwania informacji. Wyszukiwanie jednocześnie odpowiedzi na wiele pytań pozwoliłoby na jednokrotne wybieranie tych zbiorów elementarnych, które jednocześnie wchodzą w skład kilku odpowiedzi. Wykorzystanie tej możliwości zależy oczywiście od tego, jak często tego rodzaju właśnie pytania będą zadawane do systemu. Jeśli będą one niezbyt częste, możliwość wykorzystania tej własności będzie niewielka, a w konsekwencji metoda ta nie będzie się różniła od innych metod wyszukiwania z tego właśnie punktu widzenia.

Wykorzystanie ewentualnych zalet metody składowych atomowych w wielodostępnych systemach wyszukiwania wiąże się również z analizą czasu oczekiwania zbiorów elementarnych do wykorzystania w odpowiedzi. Obsługa takich kolejek wymaga odpowiedniej obsługi programowej, co z kolei powoduje zmniejszenie efektywności systemu. Tak więc celowość stosowania proponowanej metody w takim przypadku zależy od dokładnej analizy czasów obsługi użytkowników.

Zwróćmy uwagę, że stosując metodę składników atomowych w systemach wielodostępnych tworzymy kolejki termów prostych, które mogą być przydatne do odpowiedzi na wiele pytań jednocześnie, ale nie tworzymy kolejek całych pytań, jak w systemach dotychczasowych.

B6. SYSTEMY ROZPROSZONE

W przypadku systemów rozproszonych typu takiego, jaki omówiono w rozdz. 4, metoda składowych atomowych wydaje się mieć pewne zalety, pozwala ona na proste znalezienie odpowiedzi w takich systemach przez rozbicie pytania na odpowiednie składowe, na które odpowiedź jest wyszukiwana w systemach lokalnych, a następnie odpowiedzi częściowe na składane na odpowiedź całkowitą. Zauważmy, że mogą tu występować dwie sytuacje. W pierwszej mamy już istniejące systemy lokalne i chcemy je połączyć w jeden system wyszukiwania. Wtedy mamy już strukturę bazy danych i możemy jedynie stosować metodę składowych atomowych do systemu jako całości. W drugim przypadku projektowany system informacyjny jeszcze nie istnieje i mamy możliwość wyboru struktury również systemów lokalnych. Metoda składowych atomowych może w takim przypadku być stosowana zarówno na szczeblu centralnym, jak i lokalnym, o ile oczywiście ma to uzasadnienie techniczne.

Zauważmy na koniec, że idee rozproszenia bez danych można również wykorzystać w systemie, który faktycznie nie jest systemem rozproszonym, w celu przyspieszenia jego działania. Pamięć takiego systemu powinna być podzielona wtedy na segmenty odpowiadające zbiorom elementarnym lub ich zespołom i do nich mogą być zastosowane algorytmy podane w rozdz. 4. Może to być szczególnie przydatne w systemach mikrokomputerowych.

B7. KONSEKWENCJE SPRZĘTOWE

Z podanej metody wynika również architektura maszyn specjalnie przystosowanych do tej metody. Badania nad architekturą maszyn liczących z punktu widzenia organizacji baz danych oraz metod wyszukiwania są ostatnio rozwijane bardzo intensywnie. Od strony metody składowych atomowych zagadnienie to nie było jednakże do tej pory badane.

Wydaje się tu, że główną zaletą tego typu rozwiązań powinna być możliwość równoległego działania maszyny w trakcie realizacji algorytmu wyszukiwania, co w konsekwencji prowadzi do używania pamięci dzielonej na niezależne bloki oraz wyszukiwania pozwalającego na niezależną obsługę tych bloków oraz kolejek.

Zasada ta może być stosowana na różnych poziomach organizacji maszyny, a więc zarówno na poziomie pojedynczej jednostki przetwarzającej, jak i ich zespołów.

LITERATURA

1. Aho K. V., Sagiv Y., Ullman I. D.: Efficient optymization of a class of relational expressions. *ACM Tran. on Database Systems*, Dec. 1979, vol. 4, nr 4, s. 435—454.
2. Armstrong W.: Dependency structures of data base relationships. *Proc. IFIP Congress*, Stockholm, Aug. 5—10, 1974.
3. Babad J. M.: A record and file partitioning model. *Comm. ACM*, 1977, vol. 20, no 1, s. 22—31.
4. Blattner M., Brooks R., Pawlak Z.: Partitioned data structures. Waszyngton 1980 (maszynopis).
5. Błaszczyk R.: Dekompozycja atrybutowa systemów informacyjnych. Praca doktorska. Instytut Podstaw Informatyki PAN, 1982.
6. Bollmann P., Konrad E.: Fuzzy document retrieval. Interner CIS — Bericht 7/76. Informatik, Technische Universitat Berlin. Presented at the Third European Meeting on Cybernetics and System Research, Vienna, April 20—23, 1976.
7. Brooks R., Blattner M., Pawlak Z., Barret E.: Using partitioned databases for statistical data analysis. Waszyngton 1980 (maszynopis).
8. Cherniavsky V., Schneider H. J.: A data information language. Rep. No. 4/76, TU Berlin.
9. Codd E. F.: A relational model of data for large shared data banks. *Comm. ACM*, June 1970, 13, 6, s. 377—387.
10. Eagin R.: Relational data base decomposition and propositional logic. *IBM Research Report*, July-August 1976. San Jose.
11. Fila I., Wilk M.: O pewnym algorytmie klasyfikacji. Uniwersytet Śląski 1982 (maszynopis).
12. Hájek P., Havel I., Chytil M.: The GUHA method of automated hypotheses determination. *Computing*, 1966, 1. s. 293—308.
13. Hájek P., Havránek T.: On generation of inductive hypotheses. *International Journal of Man--Machine Studies*, 1977, 10, s. 3—32.
14. Hájek P., Havránek T.: *Mechanizing hypothesis formation.* Heidelberg, Springer-Verlag, 1978.
15. Hayes-Roth F.: On interference matching technique for inducting abstractions. *Comm. ACM*, 1978, vol. 21, no. 3, s. 401—411.
16. Havel I., Chaudra A. K.: Computable queries for relational data bases. *Journal of Computers and System Sciences*, 1980, 21, s. 156—178.
17. Hofter J. A.: An integer programming formulation of computer data base design problems. *Information Sciences*, 1976, 11, s. 29—48.
18. Hutt A. T. H.: A relational data base management system. New York, Wiley 1979, s. 226.
19. Imieliński T.: On some extension of the query language for incomplete information systems. *ICS PAS Report*, 1979, 349, Warszawa.
20. Imieliński T.: Note on minimization of a set of attributes. *ICS PAS Report*, 1975, 381, Warszawa.
21. Jaegermann M.: Information storage and retrieval systems — mathematical foundations. Part IV. Systems with incomplete information. *CC PAS Report*, 1975, 215, Warszawa.
22. Jaegermann M.: Incomplete storage and retrieval systems with incomplete information. Part I. *Fund. Inform.*, 1978, 2, s. 17—41.

23. Jaegermann M.: Information storage and retrieval systems with incomplete information. Part II. *Fund. Inform.*, 1979, 2, s. 141—166.
24. Jaegermann M., Lipski W.: Numerical queries in incomplete information data bases. *ICS PAS Report*, 1980, 388, Warszawa.
25. Jaegermann M., Marek W.: On dependencies of attributes in information systems. *ICS PAS Report*, 1981, 428, Warszawa.
26. Jaegermann M., Marek W., Sobolewski M.: Information storage and retrieval systems — mathematical foundations. Part III. Tree-structured-attribute systems. *CC PAS Report*, 1975, 214, Warszawa.
27. Jankowski A., Rauszer C.: Logical foundation approach to users domain restriction in data bases. *ICS PAS Report*, 1980, 418, Warszawa.
28. Kmieć K., Mucha W.: Implementacja systemu informacyjnego STUDENT realizowanego w oparciu o metodę dekompozycji. Zeszyty Naukowe Uniwersytetu Śląskiego „Systemy informacji naukowo-technicznej". Katowice 1981.
29. Koczkodaj W.: A relational model of data and its connections with the information storage and retrieval systems. *ICS PAS Report*, 1978, 306, Warszawa.
30. Konikowska B.: Continuous informational systems. *Fund. Inform.*, 1978, 2, s. 43—61. Również w: *CC PAS Report*, 1977, 273, Warszawa.
31. Konikowska B.: Constrained information systems. *ICS PAS Report*, 1977, 316, Warszawa.
32. Konikowska B., Mężyńska E., Sendova E.: A theory of information systems. *ICS PAS Report*, 1980, 421, Warszawa.
33. Konikowska B., Traczyk T.: Query language of stochastic informational systems. *Fund. Inform.*, 1979, 3, s. 351—363.
34. Konrad E., Orłowska E., Pawlak Z.: Knowledge representation systems. *ICS PAS Report*, 1981, 438, Warszawa.
35. Konrad E., Orłowska E., Pawlak Z.: On approximate concept learning. *Bericht*, October 1981, nr 81—7, TU Berlin.
36. Kuratowski K.: *Wstęp do teorii mnogości i topologii.* Warszawa, PWN 1977.
37. Lipski W.: On semantic issues with incomplete information data base. *Proc. 3rd Inf. Conf. on very Large Data Bases*, Tokyo 1977, s. 6—8 (*ACM Trans. on Database Systems*)
38. Lipski W., Jr.: On data bases with incomplete informations. *J. ACM*, 1981, vol. 28, no. 1, s. 41—70.
39. Lipski W.: Information storage and retrieval systems — mathematical foundations. Part II. *CC PAS Report* 153, Warszawa 1974.
40. Lipski W.: Information storage and retrieval — mathematical foundations. Part II. Combinatiorial problems. *Theoret. Comput. Sci.*, 1976, 3. s. 183—211.
41. Lipski W.: *Kombinatoryczne aspekty teorii wyszukiwania informacji.* Warszawa, PWN 1975.
42. Lipski W.: An efficient method of information retrieval. *CC PAS Report*, 1975, 194, Warszawa.
43. Lipski W.: On an efficient method of information retrieval. *Fund. Inform.*, 1979, 2, s. 227—243.
44. Lipski W.: Informational systems with incomplete information. *Proc. 3rd Internat. Colloq. on Automata, Languages and Programming*, Edinburgh, Scotland, 1976, Edinburgh University Press, s. 120—130.
45. Lipski W.: Informational systems: semantic issues related to incomplete information. Part. I. *CC PAS Report*, 1977, 275, Warszawa.
46. Lipski W.: On the logic of incomplete information. *Proc. 6th Internat. Symp. on Math. Foundations of Computer Science*, Tatranska Lomnica, Czechoslovakia, Sept. 1977, Springer-Verlag, 1977, s. 374—381. Również w: *ICS PAS Report*, 1977, 300, Warszawa.
47. Lipski W.: On semantic issues connected with incomplete information data bases. *ICS PAS Report*, 1977, 325, Warszawa.

48. Lipski W.: On semantic issues connected with incomplete information databases. *Proc. 3rd Internat. Conf. on Very Large Data Bases.* Part 2, Tokyo 1977, s. 71—83.
49. Lipski W.: On semantic issues connected with incomplete information databases. *ACM Trans. Database Systems,* 1979, 4, s. 262—296.
50. Lipski W.: On data bases with incomplete information. *J. ACM,* 1982, Również w: *Techn. Rep. MIT/LCS/TM-142, Laboratory for Computer Science,* MIT, Cambridge, MA, October 1979.
51. Lipski W., Marek W.: An application of graph theory to information retrieval *Bull. Acad. Polon. Sci., Ser. Sci. Math. Astronom. Phys.,* 1974, 22, s. 691—695.
52. Lipski W., Marek W.: On information storage and retrieval systems. W: *Mathematical Foundations of Computer Science,* A. Mazurkiewicz and Z. Pawlak (Eds.), Banach Center Publications, vol. 2, Warszawa, Polish Scientific Publishers, 1977, s. 215—259. Również w: *CC PAS Report* 200, Warszawa 1975.
53. Lipski W., Marek W.: Information systems: On queries involving cardinalities. *Information Systems,* 1979, 4, s. 241—246. Również w: *ICS PAS Report* 330, Warszawa 1978.
54. Lipski W., Marek W.: O sformalizowanej teorii systemów informacyjnych. *Podstawy sterowania,* 1976, 6, s. 3—22.
55. Łoś J.: Characteristic sets of a system of equivalence relations. *Colloq, Math.,* 1979, 42, s. 291—293.
56. Maeda T.: A formal treatment of document information systems. *Information Processing and Management.* vol. 17, no. 6, 1981, s. 319—328.
57. Marek W.: Some normal forms of coding and their applications to information retrieval. *CC PAS Report* 265, Warszawa 1976.
58. Marek W.: On the data base machine. *ICS PAS Report,* 1979, 347, Warszawa.
59. Marek W.: Onyszkiewicz J.: *Elementy logiki i teorii mnogości w zadaniach.* Warszawa, PWN 1975.
60. Marek W., Pawlak Z.: Mathematical foundations of information storage and retrieval. Parts 1, 2, 3, *CC PAS Reports,* 1973, 135, 136, 137, Warszawa.
61. Marek W., Pawlak Z.: On the foundations of information retrieval. *Bull. Acad. Polon. Sci., Ser. Sci. Math. Astronom. Phys.,* 1974, 22, s. 447—452.
62. Marek W., Pawlak Z.: Information storage and retrieval systems — mathematical foundations. Part I. *CC PAS Report,* 1974, 149, Warszawa.
63. Marek W., Pawlak Z.: Information storage and retrieval systems — mathematical foundations. *Theoret. Comput, Sci.* 1976, 1, s. 331—354.
64. Marek W., Pawlak Z.: Rough sets and information systems, *ICS PAS Report,* 1981, 441, Warszawa.
65. Marek W., Rode-Babczenko I.: A decomposition of informational systems. *CC PAS Report.* 1975, 212, Warszawa.
66. Marek W., Traczyk T.: Stochastic informational system. Part I. *Fund. Inform.,* 1977, 1, s. 121—130.
67. Margański E.: Procedura odwzorowania dokumentu w formacie numerycznym na numer składnika atomowego. *CC PAS Report,* 1975, 178, Warszawa.
68. Margański E.: O pewnej implementacji systemu information storage and retrieval. *CC PAS Report,* 1975, 213, Warszawa.
69. Margański E.: Organizacja i niektóre algorytmy w nowym systemie wyszukiwania. *CC PAS Report,* 1976, 267, Warszawa.
70. Margański E.: Wyszukiwanie informacji metodą składowych atomowych. Własności użytkowe *ICS PAS Report,* 1979, 363, Warszawa.
71. Margański E.: *Podstawy implementacji systemów wyszukiwania informacji metodą składowych atomowych.* Warszawa, PWN 1979.

72. Mąkosa M.: Systemy gromadzenia i wyszukiwania informacji z rozproszoną bazą danych. *CC PAS Report*, 1976, 231, Warszawa.
73. Michalewicz Z.: A note on security problem in information storage and retrieval systems. *ICS PAS Report*, 1980, 385, Warszawa.
74. Michalewicz Z.: An algorithm to compromise the data base. *ICS PAS Report*, 1980, 392, Warszawa.
75. Michalewicz Z.: Data base security. *ICS PAS Report*, 1980, 414, Warszawa.
76. Michalewicz Z.: Compromisability of a statistical data base. *ICS PAS Report*, 1980, 420, Warszawa.
77. Michalewicz Z.: A coin-weighing problem and its connection with security of a statistical database. *ICS PAS Report*, 1982, 426, Warszawa.
78. Michalski R.: Variable-valued logic and its application to pattern recognition and machine learning. 1974 (maszynopis).
79. Michalski R.: Pattern recognition as rule guided inductive inference. *IEEE Transaction on Pattern Analysis and Machine Inteligence*. July 1980, vol. PAM 1—2, no. 4, s. 349—361.
80. Michalski R. S., Chilausky R. L.: Knowledge acquisition by encoding expert rules versus computer inductive from examples: *Int. J. Man-Machine Studies*, 1980, **12**, 1, s. 63—83.
81. Novotny M.: Reducing the number of attributes in an information system. Brno 1981 (maszynopis).
82. Orłowska M.: Algebraiczne i topologiczne własności modeli baz danych systemów z niepełną informacją. Praca doktorska, Politechnika Warszawska, Warszawa 1980.
83. Orłowska E.: Dependency of attributes in information systems. *ICS PAS Report*, 1980, 425, Warszawa.
84. Orłowska E.: Dynamic information systems, *ICS PAS Report*, 1981, 434, Warszawa.
85. Orłowska E., Pawlak Z.: Expressive power of knowledge representation systems. *ICS PAS Report*, 1981, 432, Warszawa.
86. Pawlak Z.: Rough sets, basic notions. *ICS PAS Report*, 1981, 431, Warszawa.
87. Pawlak Z.: Classification of objects by means of attributes. *ICS PAS Report*, 1981, 429, Warszawa.
88. Pawlak Z.: Mathematical foundations of information retrieval. W: *Proc. Symp. on Mathematical Foundations of Computer Science*. Sept. 3—8, 1973, High Tatras. s. 135—136.
89. Pawlak Z.: Mathematical foundations of information retrieval. *CC PAS Report*, 1973, 101, Warszawa.
90. Pawlak Z.: Information systems. *ICS PAS Report*, 1979, 338, Warszawa.
91. Pawlak Z.: Distributed information systems. *ICS PAS Report*, 1979, 370, Warszawa.
92. Pawlak Z.: Toward the theory of information systems. The notion of an information system. Part I. *ICS PAS Report*, 1980, 419, Warszawa.
93. Pawlak Z.: Information systems — theoretical foundations. *Information Systems*, 1981, vol. 6, no. 3, s. 205—218.
94. Pin-Shan-Chen P., Akoka J.: Optimal design of distributed information systems. *Trans. IEEE on Computer*, Dec. 1980, vol. 6, no. 12, s. 1068—1080.
95. Rasiowa H.: *Wstęp do matematyki współczesnej*. Warszawa, PWN 1975.
96. Raś Z.: A mathematical model of a diagnostic experiment. *ICS PAS Report*, 1977, 292, Warszawa.
97. Raś Z.: Algebraiczne podstawy wyszukiwania informacji. Część I., *ICS PAS Report*, 1978, 327, Warszawa.
98. Raś Z.: Algebraiczne podstawy wyszukiwania informacji. Część II. *ICS PAS Report*, 1979, 339, Warszawa.
99. Raś Z.: Information retrieval systems — an algebraic approach. Part 1. *Proc. 1979 Conf. on Information Sciences and Systems*, Baltimore, MD, 1979, s. 419—424.

100. Raś Z.: Information retrieval systems — an algebraic approach. Part 2. *Proc. 1979 PAIS*, Durham, NC, 1979, s. 109—114.
101. Raś Z.: Information retrieval systems — an algebraic approach. Part 3. *Proc. 1980 Conf. on Information Sciences and Systems*. Princeton, NJ, 1980.
102. Sendova E.: On some classification properties of the systems with incomplete information. *ICS PAS Report*, 1980, 401, Warszawa.
103. Salton G.: Automatic information organization and retrieval. McGraw-Hill, New York 1968.
104. Stawowczyk A.: Podstawy teorii pomiarów. Praca doktorska, 1981. Politechnika Wrocławska.
105. Solak J.: O zastosowaniu minimalnych kodów w systemach wyszukiwania informacji. *ICS PAS Report*, 1980, 389, Warszawa.
106. Solak J.: O metodzie przyspieszonego wyszukiwania dokumentów. *ICS PAS Report*, 1979, 355, Warszawa.
107. Teluk S.: Dekompozycja hierarchiczna systemów informacyjnych. Praca doktorska, Instytut Podstaw Informatyki PAN, 1982.
108. Traczyk T.: Common extension of Boolean informational systems. *Fund. Inform.*, 1978, 2, s. 63—70.
109. Traczyk T.: On Pawlak information systems with limited access. *ICS PAS Report*, 1980, 387, Warszawa.
110. Truszczyński M.: Algorithmic aspects of the minimization of the set of attributes problem. *Fund. Inform.*, 1982, 4 (4).
111. Wakulicz-Deja A.: Classification of information systems. *Information Systems*, 1982.
112. Wakulicz-Deja A.: O odcinkowej metodzie wyszukiwania informacji. *Postępy Informatyki*, 1981.
113. Wakulicz-Deja A.: Odpowiadanie na pytania elementarne w trakcie przekształcania pytań. *Zeszyty Naukowe Uniwersytetu Śląskiego*, 1981.
114. Wasyluk H., Roszkowska U., Jaruga E., Doroszewski J.: Niektóre zagadnienia symptomatyki nadczynności tarczycy w ujęciu ilościowym. *Pol. Arch. Med. Wewn.*, 1973, nr 11, s. 1197—1209.
115. Wong E., Chiang T. C.: Canonical structure in attribute based file organization. *Comm. ACM*, 1971, 14, s. 593—596.
116. Zeeman E. O.: The topology of the brain and visual perception. W: M. K. Fort (ed.). *Topology of 3-manifolds and related topics*. Englewood Cliffs, N. J., Prentice-Hall 1962.
117. Zembrzuski K.: O zbiorach rozpoznających w wyszukiwaniu informacji i rozpoznawaniu obrazów. *CC PAS Report*, 1976, 239, Warszawa.
118. Ziarko W.: Systemy przechowywania i wyszukiwania informacji z niedyskretnymi zbiorami atrybutów. *CC PAS Report*, 1976, 228, Warszawa.

SKOROWIDZ

Aktualności informacji 14
alfabet języka 51
atrybut 15, 26, 33
— dziedzina 15, 16
— wartość 15
— wielowartościowy 100
— zależny od podzbioru atrybutów 28
atrybuty niezależne 26
— równoważne 26
— wielowartościowe 100
— zależne 26

Blok elementarny 21
— podziału 151
brzegowy zbiór elementarny 152

Dane o obiekcie 17
deskrypcja 17
deskryptor 17, 51, 52
— dokładny 112
— przybliżony 112
— stochastyczny 129
— wielowartościowy 108
dokładność dolnego przybliżenia zbioru 66
— górnego przybliżenia zbioru 66
— języka 64
— uczenia się 165
dokument 67
dolne przybliżenie podziału 151
— — zbioru 65, 135
dopełnienie zbioru 170
dziedzina atrybutu 15, 16
— funkcji 173
— relacji 171

Efektywność języka 66
— systemu 14

element maksymalny zbioru 172
— minimalny zbioru 172

Funkcja 172

Górne przybliżenie podziału 151
— — termu 123
— — zbioru 65, 135

Iloczyn zbiorów 170
informacja 14, 19
— niepusta 19
— o obiekcie 17
— — — w systemie stochastycznym 126
— — — — — wielowartościowym 102
— pusta 19
instrukcja 67
— liczbowa 68
— logiczna 75
— numeryczna 69
— porządkująca 77
— relacyjna 74

Język systemu 51, 132
— — centralny 88
— — dokładny 114
— — —, wartości termów 122
— — liczbowy 51
— — logiczny 51
— — lokalny 88
— — mnogościowy 51
— — numeryczny 51
— — przybliżonego 114
— — —, semantyka 115
— — —, składnia 114
— — przybliżony 114
— — relacyjny 51

Język systemu, składnia 51, 114
— — stochastyczny 128
— — wewnętrzny 117
— — wielowartościowy 108
— —, aksjomaty 108

Kardynalność odpowiedzi 68
klasy równoważności 172
klasyfikacja obiektów dwuwartościowa 151, 153
— — — niepełna 155
— — niepełna 155
— — wielowartościowa 151

Metoda składowych atomowych 10, 174
minimalna postać normalna ze względu na system 63
— próbka zbioru dolna 148
— — — górna 148
minimalny zbiór atrybutów 33
moc zbioru 170

Niepełna klasyfikacja obiektów 155
n-stopniowy system informacyjny 83

Obiekty nierozróżnialne 21, 127
— — ze względu na atrybut 21
— — — — — własność 127
— — — — — zbiór własności 127
— równoważne ze względu na atrybut 102, 112
— — — — — każdy atrybut 112
— — — — — podzbiór 102
ochrona przed niepowołanym dostępem 14
odpowiedź dokładna 81, 86
— przybliżona 81, 83, 86
ograniczenie zbioru 138
opis obiektu 17
— podzbiorów przybliżony 145
organizacja systemu 14

Pamięć centralna 96
— lokalna 96
pełność języka 60
podsystem 36, 46
— z ograniczonymi atrybutami 38
— — — obiektami 38, 67
podzbiór 170
podział 23
— jednostkowy 32
— nieopisywalny 151
podziały w przybliżeniu równe 161
— — — — od dołu 161
— — — — — góry 161
postać normalna pytań 177
— — termów 60, 61
proces uczenia 163
produkt kartezjański 171
próbka zbioru dolna 148
— — górna 148
przeciwdziedzina funkcji 173
— relacji 171
przekształcanie termów 58
przybliżenie dolne podziału 151
— — termu 123
— górne podziału 151
— — termu 123
przybliżona klasyfikacja obiektów 134
— — —, dokładność 153
— równość zbiorów 157
przybliżony opis podzbiorów 145
— — zbioru od dołu 145
— — — — góry 145
— system informacyjny 111, 112
pytanie 14, 48
— liczbowe 49
— logiczne 50
— mnogościowe 48
— numeryczne 50
—, postać normalna 49
— relacyjne 49

Redukt zbioru atrybutów 35
relacja 171
— antysymetryczna 171
— dwuczłonowa 171
— należenia do zbioru 170
— porządku 172
— porządkująca zbiór 172
— przeciwsymetryczna 171
— przeciwzwrotna 171
— równoważności 21, 172
— symetryczna 171
— tolerancji 152
reprezentacja systemu 24, 30
— złączenia 47
rozproszony system informacyjny 87
różnica zbiorów 170

Semantyka języka systemu 53
— centralnego 90, 93
stałe systemu 51, 52
stochastyczny system informacyjny 125, 126
— — —, język 128
suma zbiorów 170
system informacyjny 13, 16
— — dwustopniowy 79, 80
— — hierarchiczny 84
— — — pierwszego stopnia 85
— — kompletny 20, 35, 47, 66
— —, reprezentacja 24
— — rozproszony 87, 88, 98, 177
— — —, składowe 88
— — — z centralną pamięcią 96
— — selektywny 47, 66
— — wielowartościowy 100, 101, 105
— — —, semantyka języka 105
— — —, składnia języka 105
— — z centralnym dostępem 94
— — z rozproszonym dostępem 96
— — z rozproszonymi atrybutami 89
— — — — —, język 90
— — — — obiektami 93
— — — — —, język 93
— — zredukowany 33, 35, 46, 79, 83
— — zupełny 20, 35, 47, 66
systemy równoważne 23

Term 51, 105, 114
— charakterystyczny systemu 58
—, postać normalna 60, 61
— prawdziwy w systemie 56
— prosty 56
—, przekształcanie 58
— pusty w systemie 58
— zredukowany do zbioru atrybutów 90
termy semantycznie równe 60

Użytkownik centralny 94
— lokalny 94

Warunki zgodności 41
wartość atrybutu 15
— termu 117
— funkcji 173
wewnętrzny zbiór elementarny podziału 152
wielodostęp 177
własność 19
współczynnik dyspersji podziału 153
wyprowadzalny zbiór atrybutów 33
wyrażenie poprawne języka 52
wyszukiwanie zbiorów elementarnych 176

Zależność funkcjonalna 27
zbiór elementarny 21, 174
— — drugiego stopnia 80
— — pierwszego stopnia 80
— — podziału 152
— —, wyszukiwanie 176
— opisywalny w systemie 64
— przybliżony 135
— —, aksjomaty 140
— uporządkowany 172
zbiory w przybliżeniu równe 157
— — — — od dołu 157
— — — — od góry 157
złączenie systemów 41, 44
— atrybutami 42
— obiektami 42
znaczenie termu 93
— — przybliżone dolne 123
— — — górne 123
zredukowany system informacyjny 33, 35, 46, 79, 83