Metoda Składowych atomowych

26 stycznia 2011

Konspekt do zajęć z przedmiotu: *Systemy Wyszukiwania Informacji*

Literatura źródłowa:

1. Wakulicz-Deja A.: Podstawy systemów wyszukiwania informacji. Analiza Metod, *Problemy Współczesnej Nauki. Teoria i Zastosowania. Informatyka*, Akademicka Oficyna Wydawnicza PLJ, Warsaw, Polska, (1995)

2. Pawlak Z.: Systemy Informacyjne: podstawy teoretyczne, WNT, Warszawa, Polska, (1983)

3. Grzelak, K., Kochańska, J.: System wyszukiwania informacji metodą składowych atomowych MSAWYSZ, *ICS PAS Reports No 511*, Warszawa, (1983).

1 Główne cechy metody

Metoda zaproponowana została przez Profesora *Zdzisława Pawlaka*, który powiedział, że podstawą metody jest twierdzenie: *"każde pytanie daje się przedstawić w postaci normalnej tzn. w postaci sumy iloczynów, w których występuje tylko jeden deskryptor każdego atrybutu."*
Opracowano ją w Instytucie Matematyki Politechniki Warszawskiej.
Celem metody MSA było stworzenie możliwości ustalania licznych zbiorów atrybutów oraz uzyskiwania odpowiedzi na pytania ogólne

w krótszym czasie niż na pytania szczegółowe. Rozwiązanie takie pozwoliłoby korzystać z systemu dużej liczbie użytkowników zainteresowanych róznymi atrybutami tych samych obiektów.
System MSA został przetestowany na zbiorze prac naukowo-badawczych realizowanych w resorcie przemysłu maszynowego. Dokumenty opisujące prace charakteryzowały się dużą liczbą danych tekstowych i liczbowych.

2 Opis metody

Jest to metoda wyszukiwania informacji z grupy metod matematycznych, w której obiekty są grupowane w tzw. **składowe atomowe**. W systemie o zadanym zbiorze atrybutów A, oraz zbiorze wartości atrybutów $V_a(a \in A)$ można utworzyć dokładnie $\overset{\cup}{_{a \in A}}\ card(V_a)$ zbiorów deskryptorów (zawierających po jednym deskryptorze z każdego atrybutu). Wyznaczają one pozbiory zbioru obiektów nazwane **SKŁADOWYMI ATOMOWYMI**. Zatem w Systemie Wyszukiwania Informacji opartym na metodzie SA, odpowiedzią na pytanie jest suma składowych atomowych wyznaczonych przez deskryptory pytania doprowadzonego do postaci normalnej. Przy takim podejściu zasadniczą sprawą implementacji systemu stał się sposób przechowywania wszystkich SA w pamięci maszyny oraz metoda ich odnajdywania. W systemie zorganizowanym zgodnie z metodą SA został formalnie wprowadzony język pytań oraz określona jego semantyka. Dla systemu informacyjnego S określony jest prosty język deskryptorowy LS, który jest określony jako para **(alfabet, gramatyka)**.

3 Definicje Składowej Atomowej

- SA to najmniejsza klasa równoważności w systemie S.
- SA to zbiór obiektów będących odpowiedzią na term elementarny T_e.
- SA to zbiór obiektów nierozróżnialnych w systemie S.
- Sa to zbiór obiektów będący podzbiorem wszystkich obiektów systemu takich, że odpowiedź na term elementarny jest równa temu zbiorowi.

4 PROBLEMY IMPLEMENTACJI SYSTEMU WYSZUKIWANIA INFORMACJI METODĄ SA

Znane są następujące problemy implementacji metody MSA:

1. **Liczba zbiorów elementarnych w systemie**: Jeżeli A jest zbiorem atrybutów, a V_a zbiorem wartości atrybutu $a \in A$, to w systemie tym można utworzyć $\bigcap_{a \in A} card(V_a)$ zbiorów elementarnych. Np. dla 10 atrybutów 10 wartościowych mamy 10^{10} zbiorów elementarnych. W rzeczywistości znaczna część zbiorów jest pusta, i tylko te można uwzględnić. Wtedy MSA tylko dla mniejszej liczby atrybutów.

2. **Znalezienie efektywnej metody odnajdywania w pamięci maszyny zbiorów elementarnych**. Można numerować zbiory elementarne lub, co jest równoznaczne, numerować termy elementarne. W TA możemy przechowywać numer SA z adresem tego zbioru w pamięci.

3. **Znalezienie odpowiedniego algorytmu transformacji pytania do postaci normalnej, której wymaga metoda**. Proponowane rozwiązanie to normalizacja pytania do postaci termu elementarnego (T_e).

5 Własności Termów elementarnych (składowych atomowych):

1. $\forall_{t,t' \in T_e} \sigma(t) \cap \sigma(t') = \emptyset$ dla $t \neq t'$

2. $\cup_{t \in T_e} \sigma(t) = X$

Warunek 1 mówi, że wszystkie składowe atomowe są rozłączne a ich suma (warunek 2) jest równa pełnemu zbiorowi obiektów.
Składową atomową nazywa się zbiór $X_E \in X$ taki, że $\sigma(t_E) = X_E$, tzn. że termy elementarne opisują podział zbioru X na składowe atomowe.

6 Budowa KW

W celu stworzenia kartoteki wyszukiwawczej wprowadza się numerację deskryptorów w każdym atrybucie od 0 do $m-1$. Tak więc każdemu opisowi:

$$(a_1, v_1) * (a_2, v_2)*, ..., *(a_m, v_m)$$

, składowej atomowej będzie odpowiadał ciąg liczb:

$$b_1, b_2, ..., b_n$$

, który może być również najprostszym **identyfikatorem składowej atomowej**. Ciąg ten (zwany **ciągiem atomowym**) można także przedstawić w postaci liczby zwanej **numerem składowej atomowej** przez następujące odwzorowanie:

$$b_1, b_2, ..., b_n \Rightarrow \sum_{i=1}^{n} b_i * U_i$$

. **Liczba składowych atomowych** systemu wynosi :

$$L_{SA} = m_1 * m_2 * m_3 * ... * m_n$$

, gdzie $m_i = cardV_{a_i}$, co jest liczbą zwykle znacznie przewyższającą liczbę obiektów systemu i prowadzi do powstania znacznej liczby grup pustych.
Przykład Załóżmy, że mamy system wyszukiwania informacji w którym zbiór atrybutów składa się z 3 atrybutów $\{a, b, c\}$ i każdy z nich ma po 3 wartości np. atrybut a niech ma zbiór możliwych wartości następujący: a_1, a_2, a_3. Wówczas wartości każdego z atrybutów numerujemy liczbami naturalnymi od 0 do $m-1$ gdzie m oznacza liczbę wartości danego atrybutu. Dla atrybutu a (ale i pozostałych atrybutów b i c) deskryptorowi (a, a_1) przyporządkujemy liczbę (kod): 0, deskryptorowi (a, a_2) przyporządkujemy liczbę (kod): 1 i deskryptorowi (a, a_3) przyporządkujemy liczbę (kod): 2. W ten sposób np. opis deskryptorowy składowej atomowej równy: $(a, a_2) * (b, b_1) * (c, c_3)$ może zapisać jako ciąg atomowy (identyfikator składowej) postaci: $1, 0, 2$. Odpowiednie składowe U_i z wzoru na numer składowej atomowej tworzymy następująco:

$U_1 = m_2 * m_3 = cardV_b * cardV_c = 3 * 3 = 9$

$U_2 = m_3 = cardV_c = 3$

$U_3 = 1$

Przekształcenie ciągu atomowego na numer składowej atomowej odbywa się wówczas następująco:

$$1,0,2 \Rightarrow 1 * U_1 + 0 * U_2 + 2 * U_3 = 1 * 9 + 0 * 3 + 2 * 1 = 11$$

A więc:

$$(a, a_2) * (b, b_1) * (c, c_3) \Rightarrow 1,0,2 \Rightarrow 11$$

Ostatecznie Kartotekę Wyszukiwawczą budują:

- numer składowej atomowej (*nr SA*) bądź *ciąg atomowy* bądź *opis w postaci termu elementarnego*,
- oraz *zbiory obiektów odpowiadające danej SA*.

7 Przykład tworzenia kartoteki wyszukiwawczej

x	a	b	c
1	a1	b1	c1
2	a1	b1	c2
3	a1	b2	c1
4	a1	b2	c2
5	a2	b3	c1
6	a2	b3	c2
7	a2	b4	c1
8	a2	b4	c2

$V_a = \{a1, a2\}$
$V_b = \{b1, b2, b3, b4\}$
$V_c = \{c1, c2\}$
$L_{sa} = m_1 * m_2 * m_3$
$m_1 = cardV_a = 2$
$m_2 = cardV_b = 4$
$m_3 = cardV_c = 2$
$L_{sa} = 2 * 4 * 2 = 16$

Kartoteka wyszukiwawcza dla przykładu pokazana we wszystkich 3 postaciach: z ciągiem atomowym, numerem składowej atomowej i odpowiadającym jej zbiorem obiektów wygląda następująco:
UWAGA ! Pamiętany jest tylko jeden z następujących: ciąg atomowy,numer składowej,opis deskryptorowy oraz odpowiadający im zbiór obiektów.

opis deskryptorowy	Ciag atomowy Identyfikator SA	Nr_{sa}	Zbior obiektow
(a,a1)(b,b1)(c,c1)	000	0	X1
(a,a1)(b,b1)(c,c2)	001	1	X2
(a,a1)(b,b2)(c,c1)	010	2	X3
(a,a1)(b,b2)(c,c2)	011	3	X4
(a,a1)(b,b3)(c,c1)	020	4	$\emptyset$
(a,a1)(b,b3)(c,c2)	021	5	$\emptyset$
(a,a1)(b,b4)(c,c1)	030	6	$\emptyset$
(a,a1)(b,b4)(c,c2)	031	7	$\emptyset$
(a,a2)(b,b1)(c,c1)	100	8	$\emptyset$
(a,a2)(b,b1)(c,c2)	101	9	$\emptyset$
(a,a2)(b,b2)(c,c1)	110	10	$\emptyset$
(a,a2)(b,b2)(c,c2)	111	11	$\emptyset$
(a,a2)(b,b3)(c,c1)	120	12	X5
(a,a2)(b,b3)(c,c2)	121	13	X6
(a,a2)(b,b4)(c,c1)	130	14	X7
(a,a2)(b,b4)(c,c2)	131	15	X8

8 Wyszukiwanie odpowiedzi

Pytanie w postaci termu elementarnego, zadane do systemu zamieniamy na odpowiadający mu ciąg $b_1, b_2, ..., b_n$, zgodnie z przyjętą numeracją deskryptorów. Odpowiedzią na to pytanie będzie składowa atomowa o identycznym identyfikatorze jak wyznaczony dla pytania ciąg atomowy. W przypadku numeracji składowych atomowych odpowiedzią będzie składowa atomowa o identycznym numerze jaki został obliczony dla pytania.
W bazie danych systemu wystarczy pamiętać numery składowych atomowych i odpowiadające im zbiory obiektów.

8.1 Przykład wyszukiwania

1. pytanie jest termem elementarnym: $t \in T_E$: np. $t = (A, A2)(B, B3)(C, C1)$

   - deskryptory pytania kodujemy na przypisane im numery deskryptorów, przez co powstaje nam ciąg atomowy, który potem przekształcamy do postaci numeru Składowej Atomowej. Czyli u nas $t = (A, A2)(B, B3)(C, C1)$ zamieniamy na ciąg atomowy: 120 ponieważ deskryptor $(A, A2)$ ma przypisany kod 1, deskryptor $(B, B3)$ ma kod 2 zaś deskryptor $(C, C1)$ ma kod 0 - stąd ciąg atmowy ma postać 120.
   - Ten ciąg zamieniamy na numer Składowej Atomowej. Odbywa się to w oparciu o formułę:

     $$\sum_{i=1}^{n} b_i * U_i$$

     W naszym przypadku odpowiednie elementy tej formuły: $U_1, U_2, \ldots, U_n$ wynoszą: $U_1 = m_2 * m_3 = 4 * 2 = 8$, $U_2 = m_3 = 2$ oraz $U_3 = 1$. Przez to ciąg atomowy 120 zostanie przekształcony na $1 * 8 + 2 * 2 + 0 * 1 = 12$.
   - Odpowiedzią na pytanie t jest składowa atomowa o numerze 12 czyli
     $\sigma(t) = SA_{120} = SA_{12} = \{x_5\}$ Oczywiście składowej szukamy w całej kartotece wyszukiwawczej, a więc dokonując niejako przeglądu zupełnego.

2. pytanie nie jest termem elementarnym: $t \notin T_E$: $t = (A, A2)(B, B3)$

   - Pytanie należy znormalizować do postaci termu elementarnego. Czyli nasz term $t = (A, A2)(B, B3)$ przekształcamy jako:
     $t = (A, A2)(B, B3)[(C, C1) + (C, C2)]$
     $= (A, A2)(B, B3)(C, C1) + (A, A2)(B, B3)(C, C2)$.
   - Odpowiedź na pytanie t będzie sumą odpowiedzi na pytania składowe (będące termami elementarnymi powstałymi po normalizacji termu t). W tym przypadku będzie to suma odpowiedzi na powstałe 2 termy elementarne: t_1 i t_2.
   - Na każdy z termów szukamy odpowiedzi tak jak dla termów elementarnych, czyli przekształcamy termy na numery składowych

atomowych. W naszym przypadku będą to składowe atomowe o ciągach: 120 i 121. Te ciągi to oczywiście składowe atomowe o numerach odpowiednio: 12 oraz 13.

- $\sigma(t) = \sigma(t1) \cup \sigma(t2) = \{x_5\} \cup \{x_6\} = \{x_5, x_6\}$

9 Modyfikacje Metody Składowych Atomowych

1. **Metoda z tablicą adresową** gdzie pamiętamy tylko numer składowej atomowej (jako najbardziej optymalna reprezentacja) i dodatkowo mają to być jedynie niepuste składowe atomowe oraz zbiór obiektów odopowiadających składowej niepustej. Wówczas w procesie wyszukiwania odpowiedzi na pytania, wystarczy wyznaczyć dla zadanego pytania nr_{SA} (numer składowej atomowej) i sprawdzić w tablicy adresowej (TA) czy taki numer tam występuje. Jeśli nie ma takiego NR_{SA} w TA to odpowiedzią na pytanie t jest zbiór pusty. Ta modyfikacja pozwala nie tylko zmniejszyć zajętość pamięci potrzebną na zapamiętanie składowych atomowych i odpowiadających im zbiorów obiektów ale co równie ważne skraca to czas wyszukiwania odpowiedzi na pytania.

2. **Metoda podziału połówkowego** - gdzie dodatkowo zakładamy, że w kartotece wyszukiwawczej składowe atomowe są w pewnym porządku (leksykograficznie (dla ciągów atomowych), bądź rosnąco czy malejąco (dla numerów składowych atomowych)). W takim przypadku, wystarczy porównać pytanie (zakodowane odpowiednio na ciąg atomowy, opis deskryptorowy składowej atomowej czy wreszcie numer składowej atomowej) z rekordem będącym w połowie kartoteki wyszukiwawczej i zdecydować o tym która połowa kartoteki ma być dalej przeszukiwana. Oczywiście takiego podziału połówkowego kartoteki można dokonywać wielokrotnie wiele razy. W tej modyfikacji oczywiście nie doprowadzamy do zmniejszenia zajętości pamięci, ale znacznie skracamy czas wyszukiwania (o połowę w przypadku jednokrotnego podziału, zaś wykładniczo gdy podziału dokonujemy wielokrotnie). Co ważne, nie tracimy tym na kompletności w procesie wyszukiwania, gdyż każda składowa atomowa jest unikalna wobec czego nigdy zostawiając jakąś połowę kartoteki na rzecz innej nie ryzykujemy utratą części informacji.

3. **Metoda odcinkowa** - przyspiesza odpowiedzi na pytania ogólne. Ta metoda pozwala radzić sobie z przypadkiem, gdy pytanie jest zbyt ogólne i sprowadzanie go do postaci termu elementarnego znacznie zwiększyłoby czas wyszukiwania i ogólnie rzecz biorąc traciłoby na sensowności. Łatwiej byłoby wybrać pewien atrybut $A_i \in A$ (mający wiele wartości) i pozwolić by to on stanowił pierwsze miejsce w opisie obiektów. Następnie należy grupować obiekty w grupy (odcinki) tak, że obok siebie będą umieszczone obiekty o tej samej (j-tej) wartości atrybutu $A_i - i$ (taki blok nazywamy j-odcinkiem). Modyfikacja ta budując dodatkowo tablicę odcinków zwiększa zajętość pamięci ale pozwala skrócić w sposób znaczący czas wyszukiwania odpowiedzi na pytania zarówno ogólne jak i szczegółowe. Odcinkowanie jest tu rozumiane jako grupowanie względem od jednego do wielu atrybutów.

Wyszukiwanie:

- dla pytań elementarnych z atrybutem A_i, najpierw szukamy j-odcinka i potem w ramach wybranego j-odcinka dokonujemy przeglądu zupełnego (PZ) w celu znalezienia konkretnej składowej atomowej (SA),
- dla pytań ogólnych z atrybutem A_i, szukamy j-odcinka i cały j-odcinek jest już odpowiedzią na zadane pytanie i co ważne nie ma tu potrzeby normalizacji,
- dla pytań ogólnych ale bez atrybutu A_i, konieczna jest rzecz jasna normalizacja lub przegląd zupełny całej KW.

Przykład tworzenia kartoteki wyszukiwawczej dla metody odcinkowej

. Załóżmy, że atrybutem wielowartościowym będzie atrybutu c z czterema wartościami. Załóżmy, że kartoteka wtórna wygląda tak jak to przedstawia rysunek 1. Jeśli to atrybut c ma tworzyć kartotekę wyszukiwawczą, to powinniśmy opisy deskryptorowe (odpwiadające składowym atomowym) generować w oparciu o ten atrybut, a więc zbudowana kartoteka wtórna mogłaby mieć postać taką jak to przedstawia rysunek 2.

	a	b	c
x1	a1	b1	c1
x2	a1	b1	c2
x3	a2	b2	c3
x4	a2	b2	c4
x5	a1	b2	c1
x6	a1	b2	c2
x7	a2	b2	c3
x8	a2	b2	c4

Rysunek 1: Kartoteka wtórna

	c	a	b
x1	c1	a1	b1
x5	c1	a1	b2
x2	c2	a1	b1
x6	c2	a1	b2
x3	c3	a2	b2
x7	c3	a2	b2
x4	c4	a2	b2
x8	c4	a2	b2

Rysunek 2: Kartoteka wtórna - atrybut c na 1 miejscu

Teraz tworzymy opisy deksryptorowe zgodnie z porządkiem leksykograficznym - i przypisujemy im odpowiednie zbiory obiektów. Przedstawia to rysunek 3. Teraz pomijamy składowe atomowe puste - rysunek 4. Na koniec tworzymy tzw. **tablicę odcinków** w której pamiętamy tylko j**-odcinek** oraz **adres** danego k**-odcinka**. Przedstawia to odpowiednio rysunek 5.

Wyszukiwanie informacji w metodzie odcinkowej

Teraz jeśli już mamy tak stworzoną kartotekę wyszukiwawczą, załóżmy, że pytanie zadane do systemu ma postać:

$$t = (c, c3)$$

Wówczas szukamy deskryptora pytania w tablicy odcinów. Znajdujemy informację, że jest to 3 odcinek i adres tego odcinka wynosi: 5 a adres kolejnego odcinka wynosi 6 co oznacza, że wystarczy teraz przejść do kartoteki wyszukiwawczej w której pamiętamy składowe atomowe (rysunek 4) i znajdujemy tam informację, że odpowiada takiej składo-

opis deskryptorowy			zbiór obiektów
(c,c1)	(a,a1)	(b,b1)	x1
(c,c1)	(a,a1)	(b,b2)	x5
(c,c1)	(a,a2)	(b,b1)	ϕ
(c,c1)	(a,a2)	(b,b2)	ϕ
(c,c2)	(a,a1)	(b,b1)	x2
(c,c2)	(a,a1)	(b,b2)	x6
(c,c2)	(a,a2)	(b,b1)	ϕ
(c,c2)	(a,a2)	(b,b2)	ϕ
(c,c3)	(a,a1)	(b,b1)	ϕ
(c,c3)	(a,a1)	(b,b2)	ϕ
(c,c3)	(a,a2)	(b,b1)	ϕ
(c,c3)	(a,a2)	(b,b2)	x3,x7
(c,c4)	(a,a1)	(b,b1)	ϕ
(c,c4)	(a,a1)	(b,b2)	ϕ
(c,c4)	(a,a2)	(b,b1)	ϕ
(c,c4)	(a,a2)	(b,b2)	x4,x8

Rysunek 3: Opisy deskryptorowe i zbiory obiektów im odpowiadające

adres	opis deskryptorowy			zbiór obiektów
1	(c,c1)	(a,a1)	(b,b1)	x1
2	(c,c1)	(a,a1)	(b,b2)	x5
3	(c,c2)	(a,a1)	(b,b1)	x2
4	(c,c2)	(a,a1)	(b,b2)	x6
5	(c,c3)	(a,a2)	(b,b2)	x3,x7
6	(c,c4)	(a,a2)	(b,b2)	x4,x8

Rysunek 4: Tylko nie puste składowe atomowe

wej atomowej zbiór $\{x_3, x_7\}$. Na tym kończy się proces wyszukiwania. Warto zauważyć, że w procesie tym nie musieliśmy normalizować pytania ogólnego do postaci termu elementarnego i w tym tkwi prawdziwa zaleta tej modyfikacji.

4. **Dekompozycje: obiektowa, atrybutowa i hierarchiczne (będące treścią kolejnego rozdziału**

j-odcinek	adres j-odcinka
(c,c1)	1
(c,c2)	3
(c,c3)	5
(c,c4)	6

Rysunek 5: Tablica odcinków dla MSA

10 Dekompozycje w systemie informacyjnym

Systemy informacyjne możemy dekomponować w następujący sposób:

- w przypadku, gdy we wszystkich podsystemach zbiór obiektów jest taki sam, natomiast zbiór atrybutów jest rożny - **dekompozycja atrybutowa**,
- w przypadku, gdy we wszystkich podsystemach zbiór atrybutów jest identyczny, natomiast zbiór obiektów jest inny - **dekompozycja obiektowa**,
- w przypadku, gdy atrybuty opisujące system informacyjny pozostają w relacji zależności - **dekompozycja hierarchiczna zależna**,
- w przypadku częstego występowania w pytaniach do systemu atrybutów z pewnego podzbioru - **dekompozycja hierarchiczna wymuszona**.

10.1 Dekompozycja obiektowa

System informacyjny dekomponujemy obiektowo gdy w systemie tym operujemy na bardzo dużej liczbie obiektów opisanych tymi samymi atrybutami, przy czym liczba atrybutów jest niewielka.

10.1.1 Założenia dekompozycji obiektowej

Dla dalszych rozważań przedstawiony zostanie model dekompozycji obiektowej zgodnie z założeniami systemu funkcyjnego. Został on przyjęty, ponieważ opis obiektów oraz przyjęta metoda wyszukiwania oparte są o system

funkcyjny. System informacyjny $S =< X, A, V, q >$ dzieli się na podsystemy $S_1, S_2, ..., S_n$, gdzie:

$S_i =< X_i, A, V, q_i >$

$S = S_1 \cup S_2 \cup \ldots S_n,$

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

$X_i \subseteq X,$

$\bigcup_{i=1} X_i = X,$

$q_i = X_i \times A \rightarrow V,$

$q_i = q|X_i,$

$A_i = A$,oraz $V_i = V$.

Zbiór obiektów X w systemie S jest sumą podzbiorów obiektów X_i każdego podsystemu S_i. Zbiory atrybutów w podsystemach S_i są identyczne ze zbiorem atrybutów systemu głównego S. Funkcja informacji w podsystemach:
$q_i = X_i \times A \rightarrow V$
$q_i = q|X_i$. jest zawężeniem funkcji informacji systemu głównego wynikającym ze zmniejszonych zbiorów obiektów.
Pytanie do systemu zadawane jest w postaci normalnej:

$$t = t_1 + t_2 + t_3 + ... + t_m.$$

Odpowiedź na pytanie jest sumą odpowiedzi na pytania składowe:

$$\sigma(t) = \sigma(t_1) \cup \sigma(t_2) \cup \ldots \cup \sigma(t_m).$$

Jeżeli term elementarny t_i dotyczy wyłącznie obiektów jednego podsystemu to pytanie jest kierowane do tego podsystemu i tam znajdowana jest odpowiedź.

$$\sigma(t_i) = \sigma(t_i)|S_i = \sigma(t_i)|X_i$$

Jeżeli term elementarny t_i dotyczy obiektów z kilku podsystemów to pytanie jest kierowane do tych podsystemów, a odpowiedź otrzymujemy jako sumę odpowiedzi z podsystemów.

$$\sigma(t_i) = \bigcup_{i=1}^{m} \sigma(t_i)|S_i = \sigma(t_i)|X_i$$

gdzie m to liczba podsystemów, których dotyczy pytanie.
W każdym podsystemie znajduje się mniejsza liczba obiektów, niż w systemie centralnym. Aktualizacja przy tej dekompozycji jest prosta, natomiast redundancja nie występuje. **Modyfikacja ta zmniejsza zajętość pamięci w obrębie podsystemu, gdy w każdym z podsystemów pamiętamy tylko składowe niepuste**.

10.2 Dekompozycja atrybutowa

W systemie można dokonać dekompozycji atrybutowej gdy np. użytkowników systemu da się podzielić ze względu na rodzaj zadawanych pytań. Pytania pojedynczego użytkownika (lub grupy użytkowników), dotyczą innych grup atrybutów systemu S, lecz tego samego zbioru obiektów.

10.2.1 Założenia dekompozycji atrybutowej

Dla dalszych rozważań przedstawiony zostanie model dekompozycji atrybutowej zgodnie z założeniami systemu funkcyjnego. Został on przyjęty, ponieważ opis obiektów oraz przyjęta metoda wyszukiwania oparte są o system funkcyjny.

System informacyjny $S =< X, A, V, q >$ dzieli się na podsystemy S_1, S_2, ... , S_n, gdzie:

$S_i =< X, A_i, V, q_i >$

$S = S_1 \cup S_2 \cup \ldots S_n,$

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

$X_i = X.$

Zbiór obiektów X w każdym podsystemie jest identyczny ze zbiorem obiektów w systemie S. Zbiory atrybutów podsystemów są podzbiorami zbioru atrybutów systemu głównego S, a suma tych podzbiorów (z każdego podsystemu), daje pełny zbiór atrybutów systemu S.

$$\forall_{0<i\leqslant n} A_i \subset A \text{ tak, że } \bigcup_i A_i = A$$

Zbiory wartości atrybutów w podsystemach są podzbiorami zbioru wartości systemu S, co wynika bezpośrednio z faktu ograniczenia w podsystemach

zbioru atrybutów, a nie jest absolutnie związane z obcięciami zbiorów wartości dla poszczególnych atrybutów.

$$\forall_{0<i\leqslant n} V_i \subset V$$

Funkcja informacji w podsystemach:

$$q_i = X \times A_i \rightarrow V_i$$

$$q_i = q|X \times A_i$$

jest zawężeniem funkcji informacji systemu głównego wynikającym ze zmniejszonych zbiorów atrybutów i wartości atrybutów. W każdym podsystemie S_i systemu S jest określony inny zbiór deskryptorów D, oraz w ramach podsystemu możliwe jest stosowanie wybranej metody wyszukiwania informacji.

Pytanie do systemu zadawane jest w postaci normalnej. Odpowiedź na pytanie składowe otrzymać możemy na dwa sposoby:

1. Gdy dotyczy ono wyłącznie atrybutów z jednego podsystemu jest kierowane do tego podsystemu i tam znajdowana jest odpowiedź.

   $$\sigma(t_j) = \sigma(t_j)|S_i$$

2. Gdy atrybuty pytania składowego należą do różnych podsystemów, wówczas pytanie kierowane jest do podsystemów. Odpowiedzią końcową na term składowy jest przecięcie zbioru obiektów stanowiących odpowiedź z podsystemów.

   $$\sigma(t_j) = \sigma(t_j)|S_1 \cup \sigma(t_j)|S_2 \cup ... \cup \sigma(t_j)|S_n$$

3. Jeżeli możliwe jest określenie, do których podsystemów należy zadać pytanie, tzn. do których podsystemów należą atrybuty pytania, to odpowiedzią na nie będzie część wspólna odpowiedzi z wybranych podsystemów

   $$\sigma(t_j) = \sigma(t_j)|S_1 \cup \sigma(t_j)|S_2 \cup ... \cup \sigma(t_j)|S_m$$

   gdzie $m < n$.

W każdym podsystemie znajduje się mniejsza liczba atrybutów, niż w systemie centralnym, tym samym krótsze są opisy obiektów. Aktualizacja przy tej dekompozycji jest znacznie utrudniona, natomiast redundancja w ramach podsystemów zależy od przyjętej metody wyszukiwania informacji, a w ramach całego systemu wzrasta ze względu na występowanie obiektu w każdym podsystemie.

10.3 Dekompozycja hierarchicza

Dekompozycja ta jest jednym ze sposobów dekomponowania systemów informacyjnych. Stosuje się ją w dwóch wariantach:

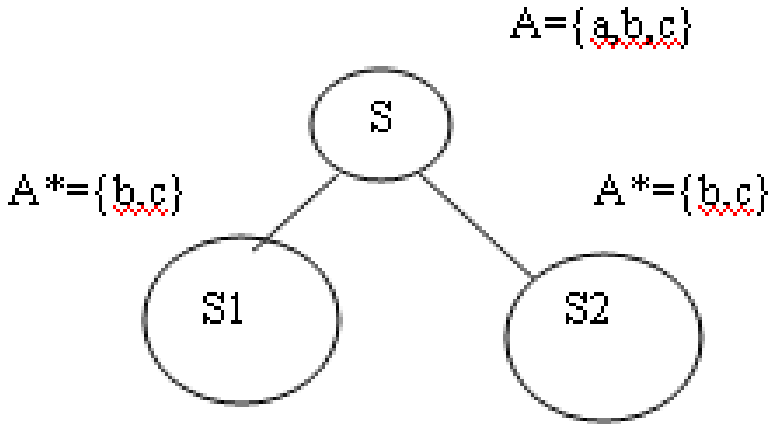
- dekompozycja hierarchiczna zależna - stosowana w przypadku gdy obiekty systemu, na którym przeprowadzona będzie ta dekompozycja są opisane przez atrybuty pozostające w relacji zależności,
- dekompozycja hierarchiczna wymuszona - dotyczy dowolnych atrybutów systemu, stosowana jest w przypadku gdy system po wstępnej eksploatacji lub analizie pytań stwierdzić może, że najczęściej w pytaniach występują razem atrybuty z pewnego podzbioru.

Dekompozycja hierarchiczna może być związana z występowaniem w systemie atrybutów zależnych lub może być wymuszona - stąd nazwy metody dekompozycji hierarchicznej: **zależna** i **wymuszona**. Przy dekompozycji hierarchicznej zależnej w systemie $S =< X, A, V, q >$ zbiór atrybutów systemu może być podzielony na dwa podzbiory A^*, B tak, że $A = A^* \cup B$, gdzie A^* jest zbiorem atrybutów niezależnych systemu S, zaś B jest zbiorem atrybutów zależnych systemu S.

x	a	b	c
1	a1	b1	c1
2	a1	b1	c2
3	a1	b2	c1
4	a1	b2	c2
5	a2	b3	c1
6	a2	b3	c2
7	a2	b4	c1
8	a2	b4	c2

$\widetilde{a} = \{\{1,2,3,4\},\{5,6,7,8\}\}$
$\widetilde{b} = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$
$\widetilde{c} = \{\{1,3,5,7\},\{2,4,6,8\}\}$
$\widetilde{b} \subset \widetilde{a} \iff b \to a$ czytamy jako: *"**a** jest zależne od **b**"*

10.4 Dekompozycja hierarchiczna zależna



Wówczas: $V1_b = \{b1, b2\}$
$V1_c = \{c1, c2\}$
$L_{sa} = 2 * 2 = 4$
System $S1$ to system:

x1	b	c
1	b1	c1
2	b1	c2
3	b2	c1
4	b2	c2

którego KW wygląda następująco:

zaś:
$V2_b = \{b3, b4\}$

Ciąg atomowy	Nr_{sa}	Zbiór obiektów
00	0	X1
01	1	X2
10	2	X3
11	3	X4

$V2_c = \{c1, c2\}$
$L_{sa} = 2 * 2 = 4$
System $S2$ to system:

x1	b	c
5	b3	c1
6	b3	c2
7	b4	c1
8	b4	c2

którego KW wygląda następująco:

Ciąg atomowy	Nr_{sa}	Zbiór obiektów
00	0	X5
01	1	X6
10	2	X7
11	3	X8

11 Łączenie systemów

Zakładając ze mamy 2 systemy S_1 i S_2:
$S1 =< X1, A1, V1, q1 >$ oraz $S2 =< X2, A2, V2, q2 >$
Powiemy, ze system S jest połączeniem systemów S_1 i S_2 ($S = S_1 \cup S_2$), gdy spełnione są następujące warunki:
$X = X_1 \cup X_2$, $A = A_1 \cup A_2$, $V = V_1 \cup V_2$, oraz dla $q|(X_1 \times A_1) = q_1$ i

$q|(X_2 \times A_2) = q_2$
To zachodzi, że: $q_x = q_{x_2} \cup q_{x_1}$ dla $x \in X$

11.1 Warunki łaczenia systemów:

1. Jeżeli $X_i \cap X_j \neq \emptyset$ oraz $A_i \cap A_j \neq \emptyset$ to:
   $q_i|_{(X_i \cap X_j) \times (A_i \cap A_j)} = q_j|_{(X_i \cap X_j) \times (A_i \cap A_j)}$ dla wszystkich $i, j = 1, 2, \ldots, n$.
2. Dla każdego $x \in X$ funkcja $q_x(q_x(a) = v)$ musi być określona dla każdego atrybutu $a \in A$: $q_x = \bigcup_i^n q_{x_i}$

Warto tu zinterpretować odpowiednio powyższe opisy:

- pierwszy warunek mówi, że jeśli w łączonych systemach są jakiekolwiek wspólne obiekty i wspólne atrybuty (o tym mówi zapis: $X_i \cap X_j \neq \emptyset$ oraz $A_i \cap A_j \neq \emptyset$) to: funkcja informacji w systemie pierwszym dla tego wspólnego obiektu i wspólnego atrybutu musi być taka sama jak w drugim systemie dla tego obiektu i atrybutu (o tym mówi zapis: $q_i|_{(X_i \cap X_j) \times (A_i \cap A_j)} = q_j|_{(X_i \cap X_j) \times (A_i \cap A_j)}$).
- drugi warunek mówi, że dla każdego obiektu $x \in X$ funkcja informacji $(q_x(a) = v)$ musi być określona dla każdego atrybutu $a \in A$: $q_x = \bigcup_i q_{x_i}$, a więc nie może być tak, że dla jakiegoś obiektu i atrybutu wspólnego dla obu łączonych systemów przypisana byłaby więcej niż jedna wartość atrybutu, albo by nie była ona znana wogóle.

11.2 Definicje dotyczące systemów i podsystemów

Zakładamy, że istnieje system funkcyjny $S =< X, A, V, q >$ oraz system $S' =< X', A', V', q' >$. Definicja podsystemu mówi, że system S' jest podsystemem systemu S, gdy zbiór $X' \subseteq X$, $A' \subseteq A$, oraz $q' = q|(X' \times A')$, $S' = S|(X' \times A')$. Prezentuje to rysunek 6.

Jeżeli dany system S' jest podsystemem systemu S oraz zbiory obiektów są identyczne $X = X'$ to powiemy, że S' jest podsystemem systemu S z ograniczonymi atrybutami $S' \subseteq S$, $S' = S|A'$. Prezentuje to rysunek 7.
Jeżeli dany system S' jest podsystemem systemu S oraz zbiory atrybutów są identyczne $A = A'$ to powiemy, że S' jest podsystemem systemu S z ograniczonymi obiektami $S' \subseteq S$ $S' = S|X'$. Prezentuje to rysunek 8.

	a	b	c	d
X1	A1	B1	C1	D1
X2	A2	B1	C2	D3
X3	A1	B2	C1	D3
X4	A1	B2	C2	D2
X5	A2	B4	C1	D1

S = <X,A,V,q>

	a	d
X1	A1	D1
X2	A2	D3
X5	A2	D1

S0 = <X0,A0,V0,q0>

Rysunek 6: System S i podsystem systemu S z ograniczonymi obiektami i atrybutami

12 Parametry Metody Składowych Atomowych

12.1 Struktura bazy danych

Struktura bazy danych jest prosta w metodzie klasycznej, staje się bardziej złożona w modyfikacjach.

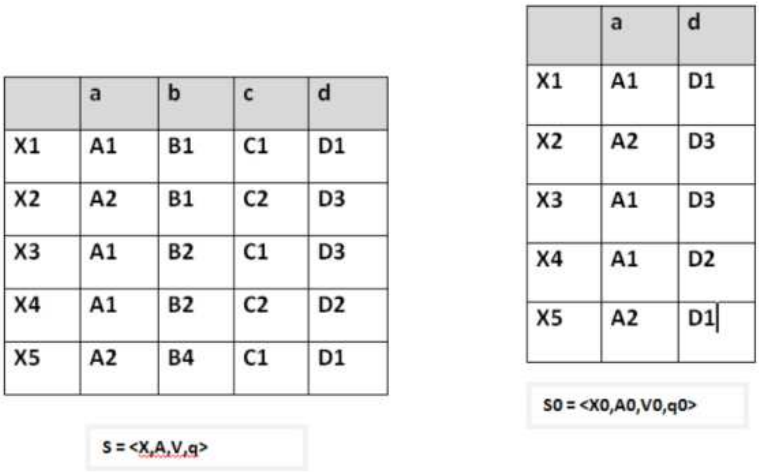
13 Redundancja i zajętość pamięci

Klasyczna metoda nie wnosi redundancji. Redundancja może wystąpić w dekompozycji obiektowej i atrybutowej. Zajętość pamięci jest największa w metodzie klasycznej i przy dekompozycji atrybutowej. W pozostałych modyfikacjach nadmiarowość jest niewielka.

14 Proces aktualizacji

Wprowadzenie nowego obiektu do bazy danych w metodzie klasycznej wymaga:

- obliczenia numeru składowej atomowej odpowiadającemu opisowi obiektu,



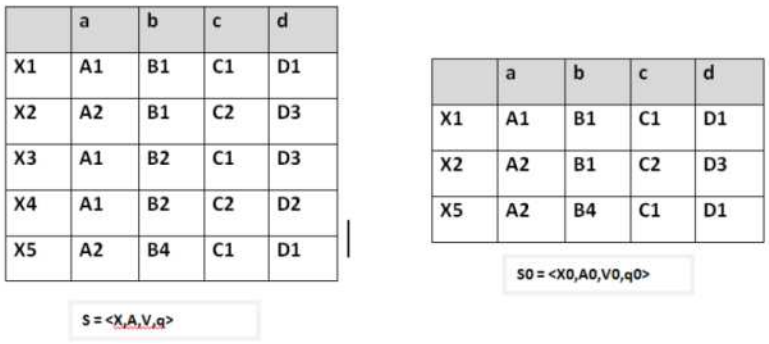
Rysunek 7: System S i podsystem systemu S z ograniczonymi atrybutami

- wprowadzenia obiektu do obliczonej składowej atomowej.

W metodzie z tablicą adresową należy sprawdzić czy obliczony numer występuje w tablicy adresowej. Jeśli **tak**, to należy dopisać obiekt pod wskazany adres, gdy **nie** ma obliczonego numeru to należy go dopisać i przyporządkować mu adres obiektu. W metodzie odcinkowej po znalezieniu odpowiedniego odcinka dopisuje się do niego nowy obiekt. W dekompozycji obiektowej obiekt należy dopisać do odpowiedniego podsystemu. W dekompozycji atrybutowej należy obiekt wpisać do każdego podsystemu. Usuwanie obiektu w metodzie klasycznej wymaga:

- obliczenia numeru składowej zgodnie z opisem obiektu,
- usunięcia obiektu z odpowiedniej składowej.

W modyfikacji z tablicą adresową należy znaleźć obliczony numer składowej w tablicy adresowej i odnajdując adres składowej usunąć z nie obiekt. Może to doprowadzić także do usunięcia składowej atomowej z tablicy adresowej.
W metodzie odcinkowej w opisie obiektu wyodrębniamy wartość wybranego atrybutu. Znajdujemy odcinek odpowiadający tej wartości i z odcinka usuwamy wskazany obiekt.
W metodzie z dekompozycją obiektową postępowanie jest jak w metodzie klasycznej tylko w odpowiednim podsystemie.



Rysunek 8: System S i podsystem systemu S z ograniczonymi obiektami

W metodzie z dekompozycją atrybutową należy usunąć obiekt we wszystkich podsystemach. Zmiany w opisie obiektu sprowadzają się do usunięcia tego obiektu i wpisania obiektu o nowym opisie.

14.1 Czas wyszukiwania

Metoda składowych atomowych ma krótsze czasy wyszukiwania odpowiedzi na pytania elementarne niż na ogólne z wyjątkiem modyfikacji odcinkowej i dekompozycji atrybutowej.

14.2 Język wyszukiwawczy

W systemie opartym na metodzie składowych atomowych stosowany jest język deskryptorowy.

14.3 Tryb pracy

Preferowany jest tryb pracy wsadowej.

15 Zadania egzaminacyjne

1. Zadanie 1. (łączenie systemów)
   Dane są następujące systemy informacyjne: $S1 =< X1, A1, V1, q1 >$ oraz $S2 =< X2, A2, V2, q2 >$, i $S3 =< X3, A3, V3, q3 >$.

(a) Podać warunki, dla których zachodzi następująca własność $S1 \cup (S2 \cup S3) = (S1 \cup S2) \cup S3$

(b) Podać przykłady systemów, które ilustrują fakt, że operacja połączenia systemów jest łączna.

Propozycja rozwiązania Warunki łączenia systemów:

(a) Jeżeli $X_1 \cap X_2 \cap X_3 \neq \emptyset$ oraz $A_1 \cap A_2 \cap A_3 \neq \emptyset$ to:
$q_1|_{(X_1 \cap X_2 \cap X_3) \times (A_1 \cap A_2 \cap A_3)}$
$q_2|_{(X_1 \cap X_2 \cap X_3) \times (A_1 \cap A_2 \cap A_3)}$
$q_3|_{(X_1 \cap X_2 \cap X_3) \times (A_1 \cap A_2 \cap A_3)}$.

(b) Dla każdego $x \in X$ funkcja $q_x(q_x(a) = v)$ musi być określona dla każdego atrybutu $a \in A$: $q_x = \bigcup_i^n q_{x_i}$

s1

a	b	c	d	e	f	
1	1	1	2	1	1	x1
1	2	1	2	2	1	x2
2	1	1	1	3	2	x3

s2

a	b	c	
2	2	2	x4
1	1	2	x5
1	2	2	x6

s3

d	e	f	
1	1	2	x4
2	2	3	x5
2	3	3	x6

(s1 u S2) u S3 = S1 u (S2 u S3)

a	b	c	d	e	f	
1	1	1	2	1	1	x1
1	2	1	2	2	1	x2
2	1	1	1	3	2	x3
2	2	2	1	1	2	x4
1	1	2	2	2	3	x5
1	2	2	2	3	3	x6

2. Obliczenie obiektów w składowych atomowych
Dany jest system $S =< X, A, V, q >$, gdzie:
$X = \{x1, x2, ..., x1000\}$
$A = \{a, b, c\}$
$Va = \{a1, a2\}$
$Vb = \{b1, b2, b3\}$
$Vc = \{c1, c2\}$
Dla powyższych założeń należy obliczyć:

- maksymalną liczbę obiektów w składowej

- przeciętną liczbę obiektów w składowej
- minimalną liczbę obiektów w składowej

Rozwiązanie
$L_{sa} = m1 * m2 * m3 = 2 * 3 * 2 = 12$

- $Max_{lsa} = 1000$ - tak będzie gdy wszystkie obiekty tworzą jedną składową.
- $Sr_{lsa} = 1000/12 = 83$
- $Min_{lsa} = 1$ Jeśli już mówimy o pewnej liczbie obiektów to mamy na myśli ich wartość dodatnią, a więc przypadek, gdy conajmniej jeden obiekt tworzy składową atomową.

3. Dany jest system informacyjny:

X	a	b	c	d
X1	V1	W1	U1	Q1
X2	V2	W2	U1	Q3
X3	V2	W2	U1	Q2
X4	V2	W2	U1	Q2
X5	V1	W2	U2	Q3

1. Wyznaczyć podziały generowane przez poszczególne atrybuty systemu.
2. Wyznaczyć zależności między atrybutami.
3. Określić własności tego systemu oraz podać ich znaczenie praktyczne.

4. Podać przykład ewidencji samochodów. Dla zapropowanego systemu określić dokładnie poszczególne elementy występujące w definicji systemu informacyjnego. Czy są w systemie obiekty nierozróżnialne? Pokazać podział na bloki elementarne. (Podać niezbędne definicje).

5. Uzasadnij następującą własność: Dla każdego podsystemu S' systemu kompletnego S z ograniczonymi atrybutami również system S' jest kompletny. Podać przykład ilustrujący tę własność.

6. Niech atrybut a systemu wyszukiwania informacji ma wartości $v1$, $v2$, $v3$. Niech dla każdego deskryptora $(a, v1)$, $(a, v2)$, $(a, v3)$ istnieje co najmniej jeden obiekt posiadający go w swoim opisie. Pokazać na drodze przekształceń formalnych, że $\sigma(t1) \neq \sigma(t2)$, gdzie $t1 = (a, v1)(\neg(a, v2))$, a $t2 = (a, v1) + (\neg(a, v2))$.

7. Omów wady i zalety dekompozycji obiektowej i dekompozycji atrybutowej w MSA.

8. Dane są dwa systemy informacyjne $S1$ i $S2$ zorganizowane zgodnie z MSA:

S1	A1	A2	A3
X1	V11	V21	V31
X2	V12	V23	V31
X3	V11	V24	V31
X4	V11	V22	V32

S2	A3	A4
X1	V31	V41
X2	V31	V42
X3	V31	V41
X4	V32	V42

   - podaj warunki jakie muszą być spełnione aby możliwe było złożenie systemów $S1$ i $S2$
   - przeanalizuj możliwości i warunki składania i zbuduj system S który jest możliwym w tym przypadku złożeniem systemów $S1$ i $S2$

9. Dla systemu S zorganizowanego zgodnie z metodą składowych atomowych

X	a	b	c
x1	v1	w1	u1
x2	v2	w1	u3
x3	v1	w2	u1
x4	v1	w2	u1
x5	v2	w2	u3
x6	v1	w1	u3

- określić język zapytań
- podać znaczenie następujących termów:
  $t1 = (a, v1) + (b, w2)(c, u2)$
  $t2 = ((a, v2)(a, v1) + (c, u3))$
  $t3 = (b, w1) \rightarrow (c, u3)$
- dla danego systemu wyznaczyć składowe niepuste i podać słowny algorytm generowania niepustych składowych atomowych.

Wskazówka:
Znaczenie termów w systemie S jest określone w następujący sposób:

(a) $\sigma(0) = \emptyset, \sigma(1) = X$,
(b) $\sigma_S(a, v) = \{x \in X : \rho_x(a) = v\}$,
(c) $\sigma_S(\sim t) = X - \sigma_S(t)$,
(d) $\sigma_S(t + t') = \sigma_S(t) \cup \sigma_S(t')$,
(e) $\sigma_S(t * t') = \sigma_S(t) \cap \sigma_S(t')$,
(f) $\sigma_S(t \rightarrow t') = \sim \sigma_S(t) \cup \sigma_S(t')$,
(g) $\sigma_S(t \leftrightarrow t') = \sigma_S(t \rightarrow t') \cap \sigma_S(t \rightarrow t')$

Zatem dla systemu danego inną przykładową tabelką:

X	a	a	c
x1	v1	w1	u2
x2	v2	w1	u3
x3	v1	w2	u1
x4	v1	w2	u1
x5	v2	w2	u3
x6	v1	w1	u3

Znaczeniem termu $(a, v_1) + (b, w_2) * (c, u_2)$ będzie zbiór $\sigma_S((a, v_1) + (b, w_2) * (c, u_2)) = \{\{x_1, x_3, x_4, x_6\} \cup \{x_3, x_4, x_5\} \cap \{x_1\}\} = \{x_1, x_3, x_4, x_6\}$. Zaś znaczeniem termu $\sim [(a, v_2) * (a, v_1)] + (c, u_3)$ będzie zbiór $\sigma_S(\sim [(a, v_2) * (a, v_1)] + (c, u_3)) = \sim (\emptyset) \cup \{x_2, x_5, x_6\} = X$.
Warto prześledzić także kolejne przykłady:

- $\sigma_S((b, w_1) + (c, u_1)) = \{x_1, x_2, x_6\} \cup \{x_3, x_4\} = \{x_1, x_2, x_3, x_4, x_6\}$.
- $\sigma_S((b, w_1) \rightarrow (c, u_3)) = (X - \{x_1, x_2, x_6\}) \cap \{x_2, x_5, x_6\} = \{x_2, x_3, x_4, x_5, x_6\}$.
- $\sigma_S((a, v_2) \leftrightarrow (b, w_2)) = \{x_1, x_3, x_4, x_5, x_6\} \cap \{x_1, x_2, x_5, x_6\} = \{x_1, x_5, x_6\}$.

10. Wadą metody składowych atomowych jest duża liczba zbiorów elementarnych. Jeżeli A jest zbiorem atrybutów oraz Va zbiorem wartości atrybutu , to w systemie można utworzyć zbiorów elementarnych. Jakie znasz sposoby rozwiązania tego problemu ?

11. Dany jest system informacyjny:
X1 = a1*b2*c3*d1*e1*f1
X2 = a2*b1*c4*d2*e1*f1
X3 = a2*b2*c3*d3*e1*f2
X4 = a3*b2*c4*d3*e1*f1
X5 = a3*b2*c4*d1*e1*f2
X6 = a1*b1*c4*d2*e1*f2

    (a) Podaj pełną definicję systemu

    (b) Przeprowadź proces normalizacji oraz podaj znaczenie termu: $t = \bar{b}2 + a1 * c2 * d1 * f3$

    (c) Czy w systemie są obiekty nierozróżnialne? (odpowiedź uzasadnić)

12. Dany jest system informacyjny $S =< X, A, V, q >$
Opisy obiektów w tym systemie są następujące:
$X1 = a1 * b2 * c3 * d8 * e1 * f1$
$X2 = a2 * b1 * c4 * d5 * e1 * f1$
$X3 = a2 * b7 * c4 * d4 * e1 * f4$
$X4 = a3 * b7 * c4 * d4 * e1 * f4$
$x5 = a5 * b2 * c4 * d8 * e1 * f5$
$x6 = a4 * b5 * c4 * d6 * e1 * f3$
$x7 = a1 * b2 * c5 * d8 * e1 * f1$
$x8 = a3 * b7 * c5 * d4 * e1 * f1$
$x9 = a2 * b7 * c5 * d4 * e1 * f3$
$x10 = a1 * b2 * c5 * d8 * e1 * f3$
gdzie: a_j oznacza j-tą wartość atrybutu a
Co to jest klasa równoważności?

Wyznaczyć klasy równoważności dla w/w systemu.
Jak można wykorzystać tę własność systemu informacyjnego?

13. Przedstaw przykładowe 3 systemy informacyjne, które da się złożyć obiektowo i atrybutowo (równocześnie). W rozwiązaniu uwzględnij polecenia:

    - Zdefiniuj poprawnie systemy funkcyjne
    - Podaj formalne warunki składania systemów (atrybutowo i obiektowo).

14. Rozpatrzmy SI 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 tabelą:

x	a	b	c
x1	p1	q1	r1
x2	p1	q2	r1
x3	p2	q3	r1
x4	p1	q1	r2
x5	p1	q2	r2
x6	p2	q3	r2
x7	p3	q1	r3
x8	p3	q2	r3
x9	p4	q3	r3

    Atrybuty definiują więc w systemie następujące podziały:
    $X_{a,p1} = \{x_1, x_2, x_4, x_5\}$
    $X_{a,p2} = \{x_3, x_6\}$
    $X_{a,p3} = \{x_7, x_8\}$
    $X_{a,p4} = \{x_9\}$
    $X_{b,q1} = \{x_1, x_4, x_7\}$
    $X_{b,q2} = \{x_2, x_5, x_8\}$
    $X_{b,q3} = \{x_3, x_6, x_9\}$
    $X_{c,r1} = \{x_1, x_2, x_3\}$
    $X_{c,r2} = \{x_4, x_5, x_6\}$

$X_{c,r3} = \{x_7, x_8, x_9\}$
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 $b, c \to a$.
Zauważmy, że jeżeli $a \to b$, to istnieje funkcja o argumentach w zbiorze wartości atrybutu a oraz przyjmująca wartości ze zbioru b, jednoznacznie przyporządkowująca wartościom atrybutu a wartości atrybutu b. Dlatego w literaturze tego typu zależności nazywa się zależnościami *funkcjonalnymi*. Jeżeli atrybut b jest zależny od atrybutu a, to istnieje funkcja f_a^b taka, że $f_a^b : V_a \to V_b$, $\rho_x(b) = f_a^b(\rho_x(a))$, wtedy i tylko wtedy, gdy $X_{b,q_x(b)} \supset X_{a,\rho_x(a)}$. 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. Dla naszego przykładu, odpowiednią funkcję $f_{b,c}^a$ przedstawiono poniżej:

a	b	c
p1	q1	r1
p1	q1	r2
p3	q1	r3
p1	q2	r1
p1	q2	r2
p3	q2	r2
p2	q3	r1
p2	q3	r2
p4	q3	r3

15. Rozpatrzmy system informacyjny określony za pomocą tabeli:

Atrybuty tego systemu generują następujące podziały:
$\widetilde{a} = \{x_1, x_2, x_5\}, \{x_3, x_4\}$
$\widetilde{b} = \{x_1\}, \{x_2, x_3, x_4, x_5\}$

x	a	b	c	d
x1	v1	w1	u1	q1
x2	v1	w2	u1	q3
x3	v2	w2	u1	q2
x4	v2	w2	u1	q2
x5	v1	w2	u2	q3

$\widetilde{c} = \{x_1, x_2, x_3, x_4\}$, $\{x_5\}$
$\widetilde{d} = \{x_1\}$, $x_3, x_4\}$, $\{x_2, x_5\}$
Podział generowany przez cały system ma postać:
$\widetilde{s} = \{x_1\}$, $\{x_2\}$, $\{x_3, x_4\}$, $\{x_5\}$
W systemie tym mamy następujące zależności między atrybutami:
$d \to b$
$d \to a$
$a, b, c \to d$
$c, d \to a, b.$

16. Rozpatrzmy złączenie dwu następujących systemów:
Złączenie obu tych systemów określa następująca tabelka:

X	a	b	c
x1	v1	u1	w2
x2	v1	u2	w1
x3	v2	u1	w2
x4	v2	u1	w2

Y	c	d	e
x3	w2	p1	q1
x4	w2	p2	q1
y1	w2	p3	q1
y2	w2	p1	q2

$X \cup Y$	a	b	c	d	e
x1	v1	u1	w2	-	-
x2	v1	u2	w1	-	-
x3	v2	u1	w2	p1	q1
x4	v2	u1	w2	p2	q1
y1	-	-	w2	p3	q1
y2	-	-	w2	p1	q2

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 możemy. Gdy zbiory atrybutów oraz zbiory obiektów w obu systemach składowych są różne i żaden z systemów składowych nie jest swoim właściwym podsystemem, wówczas złązenie takich systemów nie jest systemem informacyjnym.

17. Sprowadź do postaci normalnej term $\sim [(a, v_1) * (b, u_2)] + (b, u_1)$ dla systemu S, którego funkcja informacji przedstawia się następująco:

X	a	a	c
x1	v1	u1	w2
x2	v2	u1	w1
x3	v1	u1	w2
x4	v1	u2	w2
x5	v2	u1	w1
x6	v1	u1	w1

Reguły przekształcania termów t, p, s:

- A1.$(t + p) + s = t + (p + s)$,
  A2.$t + p = p + t$,
  A3. $t * (p + s) = t * p + t * s$,
  A4. $\sim (t + p) = \sim t * \sim p$,
  A5. $t + t = t$,
  A6. $t + 0 = t$,
  A7. $t + (t * s) = t$,
  A8. $\sim 0 = 1$,
  A9. $t + 1 = 1$,
  A10. $t + \sim t = 1$,
  A11. $\sim (\sim t) = t$.
- B1. $(t * p) * s = t * (p * s)$,
  B2. $t * p = p * t$,
  B3. $t + (p * s) = (t + p) * (t + s)$,

B4. $\sim (t * p) = \sim t + \sim p$,
B5. $t * t = t$,
B6. $t * 0 = 0$,
B7. $t * (t + s) = t$,
B8. $\sim 1 = 0$,
B9. $t * 1 = t$,
B10. $t * \ t = 0$.

- C1. $t \rightarrow p = \sim t + p$,
  C2. $t \leftrightarrow p = t * p + \sim p * \sim t$

Wówczas term $\sim [(a, v_1) * (b, u_2)] + (b, u_1)$ korzystając z reguły $B4$ można zapisać jako: $\sim (a, v_1) + \sim (b, u_2) + (b, u_1)$, co potem stosując regułę $D3$ można przekształcić do zapisu: $(a, v_2) + (b, u_1) + (b, u_1)$, który zgodnie z regułą $A5$ zapiszemy jako $(a, v_2) + (b, u_1)$, co potem zgodnie z regułą $D2$ zapiszemy jako:
$(a, v_1) + (a, v_2) = 1$
$(b, u_1) + (b, u_2) = 1$
$(c, w_1) + (c, w_2) = 1$
Zgodnie z regułą $B9$ można ten term przedstawić w postaci: $[(a, v_2) + (b, u_1)] * [(a, v_1) + (a, v_2)] * [(b, u_1) + (b, u_2)] * [(c, w_1) + (c, w_2)]$ skąd zgodnie z regułami $A3, B5, B3$ otrzymamy term:
$(a, v_1)(b, u_1)(c, w_1) +$
$(a, v_1)(b, u_1)(c, w_2) +$
$(a, v_1)(b, u_2)(c, w_1) +$
$(a, v_1)(b, u_2)(c, w_2) +$
$(a, v_2)(b, u_1)(c, w_1) +$
$(a, v_2)(b, u_2)(c, w_2)$
, który jest już w postaci normalnej.