

1. Definicja Systemu Ekspertowego:

System ekspertowy to system komputerowy zawierający w sobie wyspecjalizowaną wiedzę na temat określonego obszaru ludzkiej działalności, przy czym wiedza ta jest tak zorganizowana, że umożliwia systemowi wejście w interakcyjny dialog z użytkownikiem, w wyniku czego system może oferować rady lub podpowiadać decyzje, jak również objaśniać proces prowadzonego wnioskowania.

SE służy do rozwiązywania problemów, które charakteryzują się jedną lub wieloma z następujących cech:

- problem nie da się sformalizować w postaci liczbowej;
- cele nie dadzą się opisać za pomocą matematycznych funkcji celu;
- dane i wiedza systemu są obarczone nieznanym błędem lub są one niepełne, niepewne.

Przyczyny tworzenia systemu ekspertowego (uogólnione):

- tylko jeden (lub bardzo niewielu) specjalista posiada niezbędną wiedzę, co grozi jej utratą;
- ekspertyza jest wymagana często lub jest niezbędna w wielu miejscach;
- ekspertyza jest niezbędna w miejscach niedostępnych dla człowieka lub szkodliwych dla zdrowia.

2. Etapy tworzenia SE:

- analiza problemu, pod kątem, czy kwalifikuje się on do budowy systemu ekspertowego,
- opracowanie specyfikacji systemu, zdefiniowanie jego zadań i oczekiwanych wyników;
- przejęcie wiedzy od ekspertów i jej opracowanie;
- wybór metody reprezentacji wiedzy oraz „narzędzi” do budowy systemu;
- organizacja i kodowanie wiedzy (prototyp, pełna wersja);
- weryfikacja i testowanie systemu.

3. Architektura systemu ekspertowego.

• **Ekspert / Inżynier wiedzy:**

Inżynieria wiedzy - dziedzina sztucznej inteligencji zajmująca się projektowaniem i realizacją systemów ekspertowych. Inżynier wiedzy – projektant SE, osoba łącząca wiedzę na temat technik budowy SE z umiejętnością pozyskiwania i formalizacji wiedzy eksperckiej.

• **Moduł akwizycji wiedzy:**

Procedury pozwalające na poszerzenie zakresu wiedzy i jej modyfikację. Akwizycja wiedzy – proces pozyskiwania wiedzy niezbędnej do realizacji systemu ekspertowego. Na proces składają się: rozpoznanie problemu, wywiady z ekspertem, oraz reprezentacja wiedzy eksperta. Akwizycja kończy się w momencie zapisania wiedzy eksperta w bazie wiedzy SE.

• **Moduł objaśniający:**

Jego zasadniczym zadaniem jest wyjaśnienie strategii wnioskowania. Dzięki niemu wzrasta zaufanie użytkownika do ekspertyzy. Rodzaje objaśnień możemy podzielić na dwie główne grupy, są to:

- objaśnienia typu: Jak?
- objaśnienia typu: Dlaczego?

Moduł ten nie jest niezbędnym elementem systemu. W zastosowaniach wymagających czasu rzeczywistego, może on powodować niepotrzebne opóźnienia lub pochłaniać nadmierną ilość zasobów.

• **Interfejs użytkownika:**

Formułowanie zadań przez użytkownika poprzez procedury wejścia/wyjścia i przekazywanie wyników przez program. Jest to jedyny element systemu z którym ma bezpośredni kontakt użytkownik. Jego prawidłowa konstrukcja może decydować o sukcesie programu.

• **Mechanizm wnioskujący:**

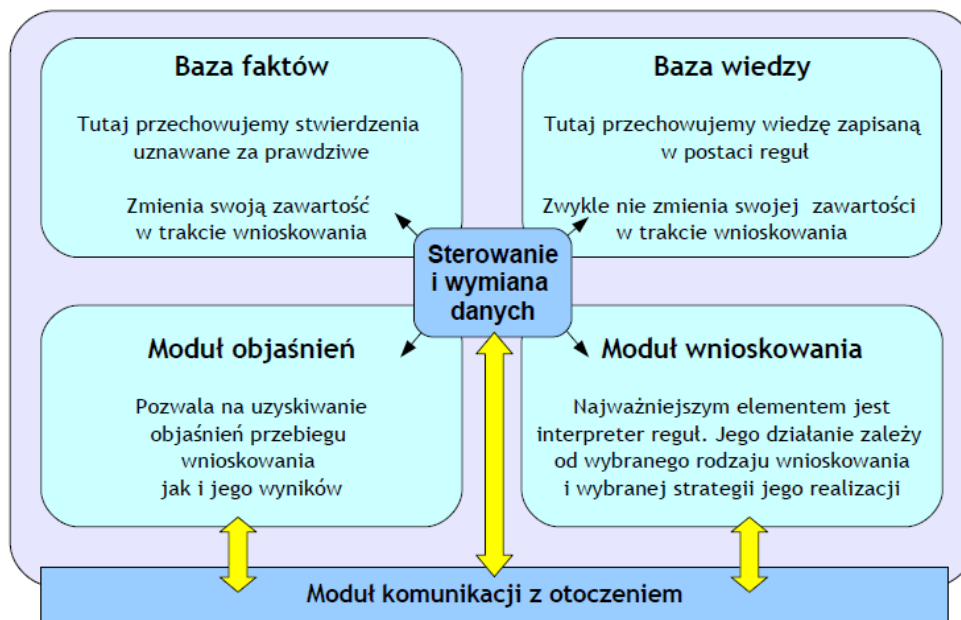
Jest to mechanizm służący do manipulowania wiedzą w oparciu o dane w celu analizy i rozwiązania zadanego przez użytkownika problemu. Składa się on z szeregu procedur przeszukiwania wiedzy i wnioskowania nowych faktów.

• **Baza wiedzy:**

Zawiera stwierdzenia oraz zasady niezbędne do rozwiązania problemów z określonej dziedziny.

Baza reguł:

W typowym systemie regułowym, stwierdzenia (fakty) są zdaniem oznajmującymi w przyjętej reprezentacji, natomiast zasady sprowadzają się do reguł postaci: IF warunek THEN wniosek AND/OR akcja.



4. Właściwości systemu ekspertowego:

- Są narzędziem kodyfikacji wiedzy,
- Mają zdolność rozwiązywania problemów specjalistycznych, w których dużą rolę odgrywa doświadczenie a wiedza ekspercka jest dobrem rzadkim i kosztownym
- Zwiększają dostępność ekspertyzy
- Poziom ekspertyzy jest stabilny – jej jakość nie zależy od warunków zewnętrznych i czasu pracy systemu
- Jawna reprezentacja wiedzy w postaci zrozumiałej dla użytkownika końcowego
- Zdolność do objaśniania znalezionych przez system rozwiązań
- Możliwość przyrostowej budowy i pielęgnacji bazy wiedzy.

5. Akwizycja wiedzy - Sposoby przejmowania wiedzy od Eksperta:

I. Ekspert sam przedstawia łańcuch logicznego rozumowania ciągnący się od przyczyny do skutku. Przekazuje swoją wiedzę w formie łańcucha przyczynowo-skutkowego typu:

„Jeśli.....to.....” .

Zalety:

- łatwo można takie łańcuchy wykorzystać do zapisu w bazie wiedzy;
- na bazie tych zapisów łatwo stworzyć blok objaśnień.

Wady:

- ekspert nie jest często w stanie przekazać wiedzy w realnym czasie;
- ekspert nie pamięta co już powiedział, a co nie;
- dla eksperta jest to zadanie nietypowe, musi przeprowadzić syntezę swojej wiedzy, a zwykle zajmuje się pojedynczymi przypadkami.

II. Ekspert określa prawdopodobieństwo wpływu pojedynczych cech (atrybutów) na podporządkowanie sytuacji określonemu zdarzeniu. (Np. lekarz-ekspert określa jakie jest prawdopodobieństwo, że taka czy inna wartość diagnozowanego objawu odpowiada pewnej chorobie. Każda para objaw-choroba rozpatrywana jest osobno.

Wady:

- ludzie z natury źle oceniają prawdopodobieństwo;
- w realnych sytuacjach, właśnie połączenia cech określają na ile ta, czy inna sytuacja jest prawdopodobna.

III. Budowa bazy wiedzy „na przykładach” (np. skorzystanie ze zdiagnozowanych przypadków (kartoteki pacjentów)). Przykłady powinny być różnorodne, ich liczba powinna być duża.

6. Akwizycja wiedzy – Pozyskanie wiedzy od wielu ekspertów:

W przypadku dysponowania ocenami wielu ekspertów trzeba opracować jednoznaczną opinię, zwaną „grupową opinią ekspertów”.

Metoda delficka – instrument pozwalający dokonać integracji ocen ekspertów.

Podstawą jest **ankieta** zawierająca pytania związane z centralnym zadaniem ekspertyzy. Sformułowania poszczególnych pytań powinny zabezpieczyć **jednoznaczność** odpowiedzi oraz wyrażenie ich w postaci ilościowej (np. punkty od 1 do 100).

W ankiecie eksperci określają źródła informacji oraz ich wpływ na ocenę.

Przebieg (zwykle wystarczają cztery rundy):

- Eksperti wypełniają kwestionariusze i otrzymują informację zwrotną o wynikach poprzedniej rundy (przy kolejnych przebiegach);
- Podają uzasadnienie wyrażonych w poprzedniej rundzie opinii;
- Otrzymują anonimowe zestawienie uzasadnień i w świetle tych uzasadnień dokonują kolejnych ocen, lub podają argumenty które wyjaśniają ich niezmiennie stanowisko.
- Proces kończymy jeśli uzyskamy pewną stabilność ocen

Mediana uzyskanych odpowiedzi jest przyjmowana (za każdym razem) jako ocena jednomyślna grupy ekspertów.

Mini metoda delficka:

- Każdy uczestnik niezależnie od innych opracowuje w wersji pisemnej swoją ocenę lub oszacowanie danej wielkości;
- Zbiór wszystkich ocen jest przedstawiany całej grupie (anonimowo);
- Krótka dyskusja nad rozbieżnościami opinii;
- Każdy uczestnik ponownie formułuje swoją ocenę;
- Mediana tych ostatnich wyników jest przyjmowana jako decyzja grupy ekspertów.

7. Rodzaje wiedzy:

• **Reprezentacja faktów:**

Podstawową formą reprezentacji dla tego rodzaju informacji są trójki <OAV>, co stanowi skrót terminu Obiekt-Atrybut-Wartość (ang. Object-Attribute-Value triples). Obiekt stanowi reprezentację pewnego podmiotu z danej dziedziny zastosowań. Obiekt zwykle posiada atrybuty go opisujące, stanowiące odzwierciedlenie cech i właściwości. Dla każdego atrybutu określa się zbiór możliwych wartości. Trójka <OAV> reprezentuje zatem informację, że dany obiekt posiada atrybut o ściśle określonej wartości.

Obiekt	: Samochód
Zbiór atrybutów	: [Kolor, Marka]
Zbiór wartości atrybutu Kolor	: [Czerwony, Zielony, Niebieski, Biały, Czarny]
Zbiór wartości atrybutu Marka	: [BMW, Volvo, Renault, Mercedes, Opel]

Przykłady trójek <OAV>:

<Samochód, Kolor, Biały>	<Samochód, Kolor, Czerwony>
<Samochód, Marka, BMW>	<Samochód, Marka, Mercedes>

- **Reprezentacja reguł (zapis wiedzy):**

Baza wiedzy w zapisie regułowym jest zbiorem reguł, które posiadają zwykle następującą postać:

Jeżeli <Przesłanka> To <Konkluzja>

lub

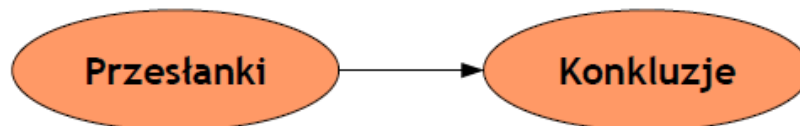
<Konkluzja> Jeżeli <Przesłanka>

gdzie:

Przesłanka: wyrażenie złożone z zdań logicznych (zwykle wyrażen prostych), połączonych spójnikami (funktorami) "and" (i) lub "or" (lub). Wyrażenie proste to zwykle trójka <OAV>, <ARV>.

Konkluzja: stwierdzenie (zdanie logiczne), które staje się prawdziwe, gdy prawdziwa jest przesłanka. Takie stwierdzenie staje się zatem faktem.

Istnieje wiele formatów zapisu reguł. Koncepcja jest jednak zwykle ta sama:



▶ Z wykorzystaniem zmiennych zdaniowych:

p – procesor się przegrzewa, q – sprawdź układ chłodzenia

$$p \rightarrow q$$

▶ Z wykorzystaniem predykatów:

$P(x)$ – procesor komp. x się przegrzewa, $Q(x)$ – sprawdź chłodzenie komp. x

$$P(x) \rightarrow Q(x)$$

▶ Z wykorzystaniem dwójek *atrybut-wartość*:

if *stan_procesora = przegrzany*
then *akcja_serwisowa = sprawdź_układ_chłodzenia*

- **Reprezentacja wiedzy niepewnej:**

Współczynnik CF związany może być z faktami, zatem trójka <O,A,W> bywa zwykle rozszerzana do czwórki <O,A,W,CF>. Również reguła może być opatrzona stopniem pewności, który mówi jak dalece ekspert jest przekonany o słuszności danej konkluzji, co bywa zapisywane:

Jeżeli <Przesłanka> To <Konkluzja> Ze Stopniem <CF>

W ogólnym przypadku współczynnik CF dla danej reguły mówi jak dalece jesteśmy przekonani, że konkluzja danej reguły jest prawdziwa w przypadku prawdziwości jej przesłanki. W najprostszym przypadku przesłanka może być np. koniunkcją pewnych faktów o ustalonym stopniu pewności. Wtedy ostateczny stopień pewności reguły wymaga przeliczenia uwzględniającego stopnie pewności faktów.

Przykład:

Dana jest następująca regułą hipotetycznej bazy wiedzy o chorobach górnych dróg oddechowych. Diagnostyce podlega pacjent, dla którego sprawdza się wystąpienie dwóch objawów choroby: gorączki i występowania bólu gardła, opisywanych przez dwuwartościowe atrybuty logiczne **Gorączka** i **BólGardła**.

Reguła:

Jeżeli Gorączka = Tak Oraz BólGardła = Tak **To** Angina **Ze** Stopniem CF = 0.7

Fakty:

<Gorączka, Tak, 0.6>

<BólGardła, Tak, 1 >

W wyniku zastosowania normalnych reguł wnioskowania otrzymamy oczywiście konkluzję, w myśl której pacjent cierpi na anginę. Ta konkluzja obarczona jest pewnym stopniem pewności. Uwzględnić on musi stopień pewności samej reguły jak i stopień pewności wystąpienia określonych faktów. Zatem ostateczny stopień pewności konkluzji $CF = 0.42$ co jest wynikiem przemnożenia stopni pewności reguły oraz faktów biorących udział w przesłance.

8. Wnioskowanie:

Wnioskowanie definiuje się na wiele różnych sposobów. Przyjmijmy, że:

- ▶ Proces wnioskowania polega na wypracowywaniu nowych stwierdzeń uznawanych za prawdziwe, opierając się na wiedzy zgromadzonej w bazie wiedzy oraz na wcześniej znanych stwierdzeniach.

Jeżeli przesłanki reguły są prawdziwe (inaczej mówiąc, są faktami) mówimy, że reguła jest spełniona i może zostać uaktywniona (odpalona). W wyniku uaktywnienia reguły, jej konkluzja staje się nowym faktem.



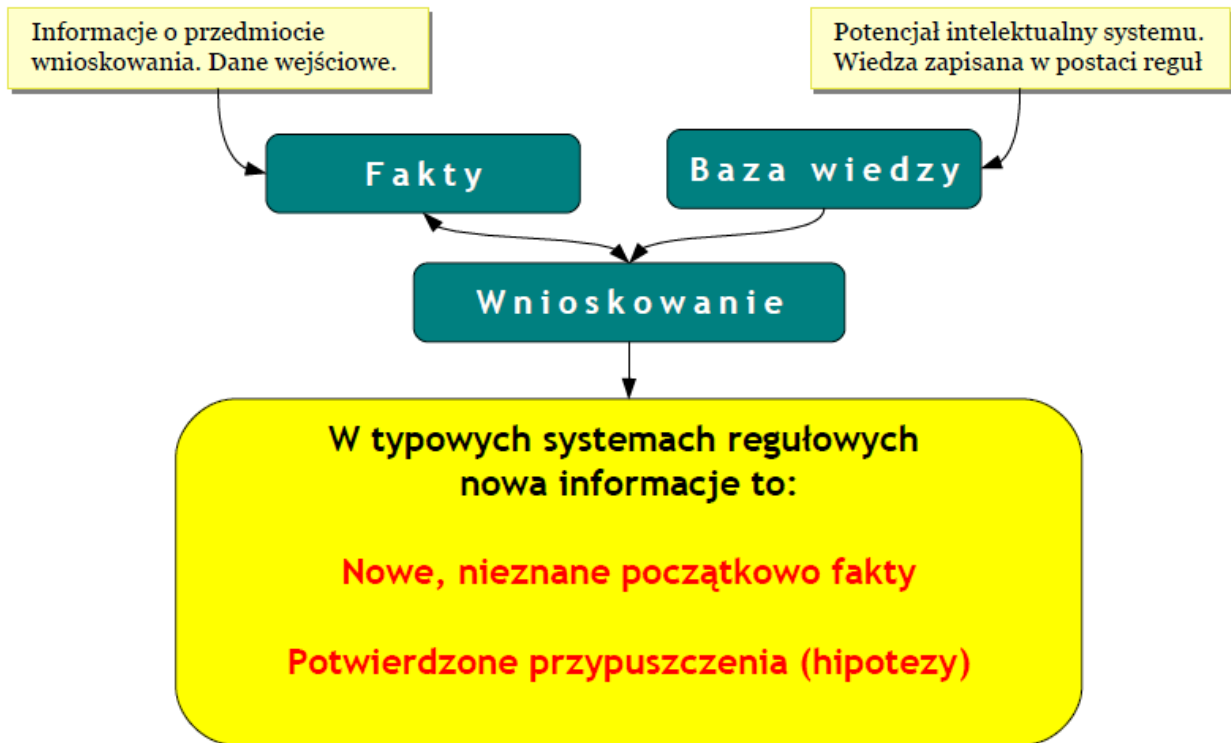
System ekspertowy – działający w oparciu o logikę dwuwartościową – w czasie wnioskowania korzysta z tzw. reguły **modus ponens**, zwanej również **regułą odrywania**.

$$\frac{A \rightarrow B, A}{B}$$

Reguła ta oznacza, że jeżeli z przesłanki A wynika B oraz A jest prawdziwe, to przyjmujemy, że fakt B jest również prawdziwy.

Dla uproszczenia Często przyjmujemy, że wystąpienie pewnego faktu w bazie wiedzy świadczy o jego prawdziwości, co znacznie przyspiesza proces wnioskowania.

Koncepcja systemu z bazą wiedzy - co możemy otrzymać?



Dwie podstawowe strategie wnioskowania

Powszechnie wykorzystuje się dwie metody wnioskowania:

- ▶ Wnioskowanie *w przód*, zwane też wnioskowaniem *progresywnym*. Polega ono na uaktywnianiu reguł spełnionych, a więc takich, których przesłanki są w zbiorze faktów. Uaktywnienie reguły powoduje dopisanie nowego faktu, co może spowodować, że spełniona i potem uaktywniona może zostać kolejna reguła.

Wnioskowanie *w przód* nie może odbyć się bez faktów. Mówi się, że jest ono **sterowane faktami** (ang. *data driven*).

Wnioskowanie *w przód* : wnioskowanie od faktów do celu (wnioskowanie sterowane danymi).

Wnioskowanie *wprzód* (idea):

Na podstawie dostępnych reguł i faktów należy generować nowe fakty tak długo, aż wśród wygenerowanych faktów znajdzie się postawiony cel (hipoteza).

Algorytm wnioskowania *w przód*

Dane wejściowe:

Zbiór reguł: $R = \{r_1, r_2, \dots, r_p, \dots, r_m\}$

Zbiór faktów: $F = \{f_1, f_2, \dots, f_p, \dots, f_n\}$

Dane robocze:

Zbiór reguł spełnionych $S \subseteq R$, tzn. takich, których przesłanki są w zbiorze F .

Zbiór reguł aktywowanych $A \subseteq R$, tzn. takich, które zostały odpalone.

Dane wyjściowe:

Zbiór F rozszerzony o nowe fakty: $f_{n+1}, f_{n+2}, \dots, f_k : F = \{f_1, f_2, \dots, f_p, \dots, f_n, f_{n+1}, f_{n+2}, \dots, f_k\}$

Algorytm:

Na podstawie F i R wyznacz zbiór S

While $S \neq \emptyset$ Do

Wybierz regułę $r_i \in S$ zgodnie z obowiązującą strategią doboru reguł

Uaktywnij regułę r_i i dopisz jej konkluzję do F

Dopisz regułę r_i do zbioru A

Na podstawie F i $R - A$ wyznacz zbiór S

Endwhile

Algorytm

Krok 1

Czy przesłanki reguł można dopasować do faktów?
Dopasowuje przesłanki reguł do faktów
Stwierdza, że **R1** oraz **R2** można uaktywnić
Na podstawie strategii „uaktywnij pierwszą” wybiera regułę **R1**
Uaktywnia regułę **R1** i wprowadza **s** do bazy wiedzy
Zapisuje, że reguła **R1** była wykorzystana

Krok 2

Dopasowuje przesłanki reguł do faktów
Stwierdza, że reguły **R2** oraz **R4** można uaktywnić
Na podstawie strategii „uaktywnij pierwszą” wybiera regułę **R2**
Uaktywnia regułę **R2** i wprowadza **t** do bazy wiedzy
Zapisuje, że reguła **R2** była wykorzystana

Krok 3

Stwierdza, że reguły **R3** i **R4** można uaktywnić
Wybiera i uaktywnia regułą **R3**, wprowadza **u** do bazy wiedzy
Zapisuje, że reguła **R3** była wykorzystana

Krok 4

Stwierdza, że regułą **R4** można uaktywnić
Uaktywnia regułą **R4**, wprowadza do bazy wiedzy **v**
Zapisuje, że reguła **R4** była wykorzystana

Krok 5

Stwierdza, że nie może uaktywnić więcej reguł
Koniec

Baza wiedzy

R1 IF p AND q THEN s
R2 IF r THEN t
R3 IF s AND t THEN u
R4 IF s AND r THEN v
p q r
+
s

R1; R2; R3; R4;

p q r s
+
t

R1; R2; R3; R4;

p q r s t
+
u

R1; R2; R3; R4;

p q r s t u
+
v

R1; R2; R3; R4;

p q r
s t u v

- Wnioskowanie *wstecz*, zwane też *regresywnym*. Polega ono na potwierdzeniu prawdziwości postawionej *hipotezy*, zwanej *celem wnioskowania*. Hipoteza jest potwierdzona wtedy, gdy istnieje reguła, której przesłanki są w bazie faktów a konkluzja zgodna jest z hipotezą. Ustalenie prawdziwości przesłanek może powodować konieczność uaktywnienia wielu reguł.

Wnioskowanie wstecz nie może odbyć się bez ustalonej hipotezy, stanowiącej cel wnioskowania. Mówi się, że jest ono **sterowane celem** (ang. *goal driven*).

Wnioskowanie wstecz (idea):

polega ono na wykazaniu prawdziwości hipotezy głównej na podstawie prawdziwości przesłanek. Jeśli nie wiemy, czy jakaś przesłanka jest prawdziwa, to traktujemy tę przesłankę jako nową hipotezę i próbujemy ją wykazać. Jeżeli w wyniku takiego postępowania zostanie znaleziona reguła, której wszystkie przesłanki są prawdziwe, to konkluzja tej reguły jest prawdziwa. Na podstawie tej konkluzji dowodzi się następną regułą, której przesłanka nie była poprzednio znana itd. Postawiona hipoteza jest prawdziwa, jeśli wszystkie rozważane przesłanki dadzą się wykazać.

- Wnioskowanie w tył : wnioskowanie od celu do faktów (wnioskowanie sterowane celem).

Algorytm wnioskowania wstecz

Dane wejściowe:

Zbiór reguł: $R = \{r_1, r_2, \dots, r_p, \dots, r_m\}$

Zbiór faktów: $F = \{f_1, f_2, \dots, f_p, \dots, f_n\}$

Cel: g

Dane robocze:

Zbiór reguł konkurencyjnych $S \subseteq R$, tzn. takich, których konkluzje są zgodne z g .

Zbiór reguł aktywowanych $A \subseteq R$, tzn. takich, które zostały odpalone.

Dane wyjściowe:

Czy $g \in F$?

Algorytm wnioskowania wstecz

Function WnioskowanieWstecz(g) : Boolean;

Begin

If $g \in F$ **Then**

Hipoteza g jest potwierdzona

return True

Else

Na podstawie g i R wyznacz zbiór S

Repeat

Wybierz regułę $r_i \in S$ zgodnie z obowiązującą strategią doboru reguł

$przeslankaPrawdziwa = \text{True}$

Foreach przeslanka p reguły r_i **Do**

$przeslankaPrawdziwa := (p \in F)$

If Not $przeslankaPrawdziwa$ **Then**

$przeslankaPrawdziwa = \text{WnioskowanieWstecz}(p)$

Until $przeslankaPrawdziwa$ **Or** $S = \emptyset$

return $przeslankaPrawdziwa$

End

Mechanizm wnioskowania wstecz

(dowodzimy: v)

Algorytm

Krok 1

Sprawdza, czy v jest zawarte w bazie wiedzy
Stwierdza, że v nie występuje w bazie wiedzy
Postuluje v i próbuje wykazać v w następnym kroku

Baza wiedzy

R1 IF p AND q THEN s
R2 IF r THEN t
R3 IF s AND t THEN u
R4 IF s AND r THEN v
 p ; q ; r ;

Krok 2

Sprawdza, czy v da się dopasować do konkluzji reguł
Stwierdza, że reguła **R4** ma v jako konkluzję
Sprawdza, czy pierwsza przesłanka reguły **R4** jest spełniona (tzn. czy s jest w bazie wiedzy)
Stwierdza, że s nie jest w bazie wiedzy
Postuluje s i próbuje wykazać s

v – cel dowodzenia

s – cel pomocniczy

Krok 3

Sprawdza, czy s da się dopasować do konkluzji reguł
Stwierdza, że reguła **R1** ma s jako konkluzję
Sprawdza, czy pierwsza przesłanka reguły **R1** jest spełniona
Stwierdza, że p jest w bazie wiedzy
Sprawdza, czy druga przesłanka reguły **R1** jest spełniona
Stwierdza, że q jest w bazie wiedzy
Stwierdza, że wszystkie przesłanki reguły **R1** są spełnione
Uaktywnia regułę **R1** i wprowadza s do bazy wiedzy

p – cel pomocniczy

q – cel pomocniczy

BW := BW + s

Krok 4

Jako kontynuacja kroku 2 jest sprawdzane czy druga przesłanka reguły **R4** jest spełniona
Stwierdza, że r jest w bazie wiedzy
Stwierdza, że wszystkie przesłanki reguły **R4** są spełnione
Uaktywnia regułę **R4** i wprowadza v do bazy wiedzy

r – cel pomocniczy

BW := BW + v

Krok 5

Jako kontynuacja kroku 1 konkluduje, że hipoteza v została wykazana
Koniec

R1; R2; R3; R4
 p q r
 s , v

Wnioskowanie mieszane (idea)

Strategia wnioskowania mieszane opiera się na wykorzystaniu ogólnych reguł, tzw. *metareguł* stanowiących metawiedzę, na podstawie której program zarządzający dokonuje odpowiedniego przełączania między wnioskowanie wstecz i wprzód.

Sterowanie wnioskowaniem (strategie)

Strategia świeżości polega na określeniu reguły, która spośród wybranych do oceny została najpóźniej dołączona do pewnego obszaru pamięci, gdzie są przechowywane reguły.

Strategia blokowania ma za zadanie eliminować te reguły, które wcześniej w procesie wnioskowania były już wykorzystane.

Strategia specyficzności opiera swoje działanie na uwzględnianiu różnej liczby przesłanek w regułach.

Metody sterowania wnioskowaniem pozwalają zastosować określoną regułę z możliwych do uaktywnienia. Stosuje się strategie, które ograniczają liczbę reguł możliwych do uaktywnienia.

- **Strategia świeżości**
polega na określeniu reguły, która spośród wybranych do oceny została najpóźniej dołączona do pewnego obszaru pamięci, gdzie są przechowywane reguły. Inny sposób polega na wyborze reguły, która najpóźniej została dołączona do reguł znajdujących się w pewnym wyszczególnionym obszarze pamięci komputera (tzw. agenda).
- **Strategia blokowania**
eliminuje reguły wykorzystane wcześniej w procesie wnioskowania. Jest to konieczne, gdyż inaczej powstałaby pętla nieskończona. Blokowanie reguł osiąga się np. umieszczając dodatkowe pole w strukturze danych (rekordzie) każdej reguły. W polu tym znajdzie się informacja o wykorzystaniu bądź niewykorzystaniu danej reguły.
- **Strategia specyficzności**
działa uwzględniając różnej liczby przesłanek w regułach. Są preferowane te reguły, które mają większą liczbę przesłanek. W przypadku reguł o tej samej liczbie przesłanek, jest wybierana ta, która ma mniejszą liczbę zmiennych.

Strategie te spełniają w programie funkcję pewnego rodzaju filtrów, mają za zadanie ograniczyć liczbę reguł - kandydatów do wykorzystania - tak, aby wybrać tylko jedną. Jeśli ciągle istnieje więcej niż jedna reguła do uaktywnienia, to stosuje się tzw. strategię przypadkowości, która wybiera regułę w sposób losowy.

Pytania i odpowiedzi:

- Który algorytm jest trudniejszy w implementacji?
W tył, ponieważ idziemy od celu do faktu i jest to algorytm rekurencyjny. Przykładem jest to, że jeśli nie wiemy, czy jakaś przesłanka jest prawdziwa, to traktujemy tę przesłankę jako nową hipotezę i próbujemy ją wykazać. Z tego wynika, że wykonujemy rekurencję ponieważ próbujemy wykazać nową hipotezę, która pomoże wykazać główną hipotezę.
- Który generuje więcej faktów?
W przód, ponieważ podczas wnioskowania uaktywniamy wszystkie reguły, które mogą nam pomóc w wykazaniu głównego celu (hipotezy). Zależy to również od strategii wnioskowania.
- Jakie prawo logiczne jest stosowane we wnioskowaniu w przód?

System ekspertowy – działający w oparciu o logikę dwuwartościową – w czasie wnioskowania korzysta z tzw. reguły **modus ponens**, zwanej również **regułą odrywania**.

$$\frac{A \rightarrow B, A}{B}$$

Reguła ta oznacza, że jeżeli z przesłanki A wynika B oraz A jest prawdziwe, to przyjmujemy, że fakt B jest również prawdziwy.

- Co jeżeli cel jest w zbiorze faktów?
Często przyjmujemy, że wystąpienie pewnego faktu w bazie wiedzy świadczy o jego prawdziwości. Nawiązując do tego – jeśli cel jest w bazie faktów – tzn. że jest prawdziwy, więc tak jakby udowodniona została postawiona hipoteza.

9. Teoria zbiorów przybliżonych

9.1. System informacyjny:

Formalnie systemem informacyjnym nazywamy uporządkowaną czwórkę:

$$SI = (U, A, V, f)$$

gdzie:

- U jest niepustym, skończonym zbiorem zwanym uniwersum, przy czym elementy zbioru, U nazywamy obiektami $U = \{x_1, x_2, \dots, x_n\}$
- A jest niepustym, skończonym zbiorem atrybutów:
 $U = \{a_1, a_2, \dots, a_m\}$

- V jest zbiorem wartości atrybutów ze zbioru A :

$$V = \bigcup_{a \in A} V_a$$

przy czym V_a nazywamy dziedziną atrybutu $a \in A$.

- $f : U \times A \rightarrow V$ jest funkcją informacji taką, że

$$\forall_{\substack{x \in U \\ a \in A}} f(x, a) \in V_a$$

9.2. System informacyjny a decyzyjny:

33. Różnice między systemem informacyjnym, a decyzyjnym

SI = $\langle X, A, V, \rho \rangle$ - system informacyjny
 X – skończony, niepusty zbiór obiektów
 A - skończony, niepusty zbiór atrybutów
 V – zbiór wartości atrybutów
 f – funkcja informacyjna
 $f: X \times A \rightarrow V$

SD = $\langle X, C, D, V, f \rangle$ - system decyzyjny
 X – skończony, niepusty zbiór obiektów
 C - skończony zbiór atrybutów warunkowych,
 $C = \{C_1, \dots, C_n\}$
 D - skończony zbiór atrybutów decyzyjnych,
 $D = \{D_1, \dots, D_n\}$
 V – zbiór wartości atrybutów
 f - funkcja decyzyjna,
 $f: X \times C \rightarrow V_D$

9.3. Przykłady:

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
5	tak	nie	wysoka	nie
6	nie	tak	normalna	nie

System informacyjny

$$SI = (U, A, V, f)$$

tablica decyzyjna

$$SI = (U, A \cup \{d\}, V, f)$$

Rozpatrywany system informacyjny może zostać zapisany w następującej postaci: $SI = (U, A, V, f)$ gdzie:

- $U = \{1, 2, 3, 4, 5, 6\}$
- $A = \{\text{Ból głowy, Ból mięśni, Temperatura, Grypa}\}$
- $V = V_{\text{Błgłowy}} \cup V_{\text{Błmini}} \cup V_{\text{Temperatura}} \cup V_{\text{Grypa}}$
 $V_{\text{Błgłowy}} = \{\text{nie, tak}\}$
 $V_{\text{Błmini}} = \{\text{nie, tak}\}$
 $V_{\text{Temperatura}} = \{\text{normalna, wysoka, bardzo wysoka}\}$
 $V_{\text{Grypa}} = \{\text{nie, tak}\}$
- $f : U \times A \rightarrow V$ (np. $f(1, \text{Ból głowy}) = \text{nie}$; $f(3, \text{Grypa}) = \text{tak}$)

Atrybuty zawarte w zbiorze A są nazywane *warunkowymi* albo po prostu *warunkami*, zaś d jest nazywane *konkluzją* bądź po prostu *decyzją systemu*. Zbiory te są zbiorami skończonymi.

i -ta **klasa decyzyjna** to zbiór obiektów $C_i = \{x \in U : d(x) = d_i\}$, gdzie d_i jest i -tą wartością decyzji odpowiadającą zbiorowi wartości decyzji

$V_d = \{d_1, \dots, d_{|V_d|}\}$.

Reguła decyzyjna jest formułą: $(a_{i_1} = v_1) \wedge \dots \wedge (a_{i_k} = v_k) \Rightarrow (d = v_d)$, gdzie $1 \leq i_1 < \dots < i_k \leq m, v_j \in V_{a_{i_j}}$.

9.4. Iloczyn Kartezjański - przypomnienie:

Powtórka - iloczyn kartezjański

Iloczyn kartezjański zbiorów A i B to zbiór wszystkich par uporządkowanych (a, b) , takich, że a należy do zbioru A , zaś b należy do zbioru B . Oznacza się go symbolem $A \times B$. Formalnie:

$$A \times B = \{(a, b) : a \in A, b \in B\}$$

Iloczyn kartezjański może być zbudowany na tym samym zbiorze, np. $A \times A$, co bywa oznaczane A^2 . Formalnie:

$$A \times A = \{(a, b) : a \in A, b \in A\}$$

Iloczyn kartezjański dla zbioru obiektów U tablicy decyzyjnej DT

U	A			
	a	b	c	d
1	1	0	2	2
2	0	1	1	2
3	0	0	2	2
4	2	2	1	0

Iloczyn kartezjański $U \times U$ to zbiór par obiektów.

$$U \times U = \{(x, y) : x \in U, y \in U\}$$

$$U \times U = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}$$

Powtórka - relacja

- ▶ *Relacja* pomiędzy elementami zbioru A a elementami zbioru B to wybrany podzbiór iloczynu kartezjańskiego $A \times B$. Relację tworzą pary elementów wybrane z iloczynu kartezjańskiego według pewnego kryterium.
- ▶ W praktyce najpopularniejsze i najszerzej stosowane są *relacje dwuargumentowe* (dwuczłonowe, binarne), zwykle nazywane po prostu relacjami.
- ▶ Jeśli założymy, że relacja nazywa się np. R , to zapis $x R y$ oznacza, że x jest w relacji R z y .

9.5. Relacja nierozróżnialności:

Relacja nierozróżnialności

O relacji nierozróżnialności mówimy wówczas, gdy w rozpatrywanym systemie mamy do czynienia z obiektami o identycznych opisach, bądź obiektami o tej samej wartości danego atrybutu (-ów [kilku, nie wszystkich]).

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
5	tak	nie	wysoka	nie
6	nie	tak	normalna	nie

Analizując poszczególne obiekty z tabeli można zaobserwować, że obiekty o numerach 1, 4 i 6 mają te same wartości atrybutów: ból głowy oraz ból mięśni zaś obiekty o numerach 1 i 5 mają tę samą wartość atrybutu temperatura. O obiektach numer 1, 4 i 6 powiemy, że są **nierozróżnialne ze względu na atrybuty: ból głowy oraz ból mięśni**, zaś obiekty o numerach 1 i 5 są **nierozróżnialne ze względu na atrybut: temperatura**.

Definicja

Niech $SI = (U, A)$ będzie systemem informacyjnym i niech $B \subseteq A$.

W zbiorze U definiujemy dwuargumentową relację $IND(B)$, generowaną przez zbiór B , zwaną relacją *nierozróżnialności* (ang. *indiscernibility relation*):

$$IND(B) = \{(x, y) \in U \times U : \forall a \in B, a(x) = a(y)\}$$

gdzie znak „=” między $a(x)$ i $a(y)$ należy rozumieć w ten sposób, że dla obiektów x i y , należących do U , atrybut a przyjmuje taką samą wartość.

Zapis w postaci:

$$x \text{ } IND(B) \text{ } y$$

Oznacza, że x jest w relacji $IND(B)$ z y .

Mówiąc konkretnie: obiekt x systemu informacyjnego SI jest nierozróżnialny od obiektu y tego samego systemu, ze względu na wybrany podzbiór atrybutów B .

Dla systemu informacyjnego przedstawionego w tabeli można wyznaczyć relacje nierozróżnialności generowane przez różne zbiory atrybutów:

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
5	tak	nie	wysoka	nie
6	nie	tak	normalna	nie

Tablica: System informacyjny / tablica decyzyjna

Niech: $A_1 = \{g, m, t\}$, $A_2 = \{t\}$, $A_3 = \{g, m\}$, $A_4 = \{g, t, c\}$, $A_5 = \{g, m, t, c\}$

$IND_{SI}(A_1) = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (2, 5), (5, 2)\}$

$IND_{SI}(A_2) = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (1, 2), (2, 1), (1, 5), (5, 1), (2, 5), (5, 2), (3, 4), (4, 3)\}$

$IND_{SI}(A_3) = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (1, 4), (4, 1), (1, 6), (6, 1), (4, 6), (6, 4), (2, 5), (5, 2)\}$

$IND_{SI}(A_4) = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)\}$

9.6. Właściwości relacji nierozróżnialności:

Relacja nierozróżnialności $IND(B)$ jest relacją równoważnościową, spełniającą następujące warunki:

1. Jest relacją zwrotną

$$\forall x \in U (x \text{ } IND(B) \text{ } x) \quad , \text{ bo:}$$

Weźmy dowolny obiekt $x \in U$, mamy więc: $\forall a \in B, a(x) = a(x)$

a więc z definicji: $(x, x) \in IND(B)$.

2. Jest relacją symetryczną

$$\forall x, y \in U (x \text{ IND } (B) y) \Leftrightarrow (y \text{ IND } (B) x)$$

Weźmy dowolne obiekty $x, y \in U$, załóżmy, że $(x, y) \in \text{IND}(B)$, mamy wtedy:

$$\forall a \in B, a(x) = a(y)$$

stąd: $\forall a \in B, a(y) = a(x)$

a więc: $(y, x) \in \text{IND}(B)$.

3. Jest relacją przechodnią

$$\forall x, y, z \in U ((x \text{ IND } (B) y) \wedge (y \text{ IND } (B) z)) \Rightarrow (x \text{ IND } (B) z)$$

Weźmy dowolne obiekty $x, y, z \in U$, załóżmy, że $(x, y) \in \text{IND}(B)$ oraz $(y, z) \in \text{IND}(B)$, mamy wtedy:

$$\forall a \in B, (a(x) = a(y) \wedge a(y) = a(z))$$

stąd:

$$\forall a \in B, a(x) = a(z)$$

a więc:

$$(x, z) \in \text{IND}(B)$$

9.7. Klasy abstrakcji:

Relacja nierozróżnialności $\text{IND}(B)$ będąc relacją równoważnościową, dzieli zbiór obiektów U na rozłączne, niepuste klasy abstrakcji. Klasy abstrakcji relacji nierozróżnialności $\text{IND}(B)$ oznacza się $U/\text{IND}(B)$.

Każda klasa abstrakcji relacji nierozróżnialności $\text{IND}(B)$ to zbiór obiektów nierozróżnialnych ze względu na atrybuty ze zbioru B .

Klasy abstrakcji $U/\text{IND}(B)$ relacji nierozróżnialności $\text{IND}(B)$ to zatem zbiór zbiorów takich obiektów, które są nierozróżnialne ze względu na atrybuty ze zbioru B .

Klasa abstrakcji dla obiektu $x \in U$ relacji $\text{IND}(B)$ wygląda następująco:

$$[x]_{\text{IND}(B)} = \{y \in U, \forall a \in B, (a(x) = a(y))\}$$

U	A			
	a	b	c	d
1	1	0	2	2
2	0	1	1	2
3	0	0	2	2
4	2	2	1	0

Dla $B = \{a\}$: $U/\text{IND}(B) = \{\{1\}, \{2,3\}, \{4\}\}$

Dla $B = \{b, c\}$: $U/\text{IND}(B) = \{\{1, 3\}, \{2\}, \{4\}\}$

Dla $B = \{a, b, c\}$: $U/\text{IND}(B) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$

Dla $B = \{c\}$ i $x=1$: $[x]_{\text{IND}(B)} = \{1,3\}$

Dla $B = \{c\}$ i $x=2$: $[x]_{\text{IND}(B)} = \{2,4\}$

Dla $B = \{a, b\}$ i $x=1$: $[x]_{\text{IND}(B)} = \{1\}$

9.8. Zbiory przybliżone:

Jednym z celów wnioskowania w systemach decyzyjnych jest próba stwierdzenia czy obiekt należy do pewnej klasy, czy nie. Obiekty posiadające identyczne opisy, lecz zaliczone do różnych pojęć, uniemożliwiają stworzenie jednoznacznej definicji tychże pojęć. Problem ten nazywamy inaczej niespójnością danych. Ważnym problemem jest zdolność radzenia sobie z takimi (niedoskonałymi) danymi.

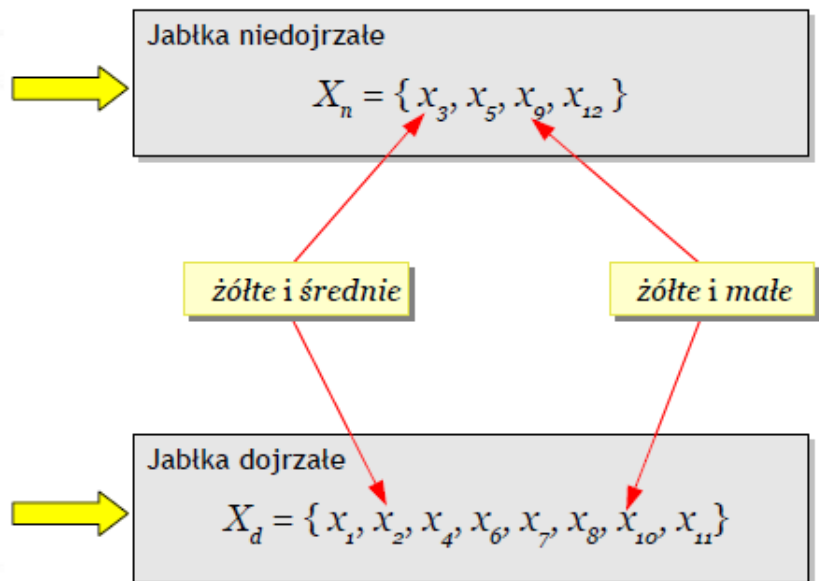
Teoria zbiorów przybliżonych zaproponowana przez Zdzisława Pawłaka jest dogodnym narzędziem analizy tego typu niespójności informacji. W przypadku niemożliwości precyzyjnego zdefiniowania zbioru obiektów (pojęcia, klasy decyzyjnej) teoria ta tworzy dolne i górne przybliżenie zbioru na podstawie klas relacji nierozróżnialności pomiędzy obiektami. Nazywamy to aproksymacją (przybliżeniem zbiorów).

(Teoria ta proponuje zastąpienie nieostrego (nieprecyzyjnego) pojęcia, parą pojęć precyzyjnych, zwanych dolnym i górnym przybliżeniem tego pojęcia. Różnica między górnym i dolnym przybliżeniem jest właśnie tym obszarem granicznym, do którego należą wszystkie przypadki, które nie mogą być prawidłowo zaklasyfikowane na podstawie aktualnej wiedzy. Im większy obszar graniczny pojęcia tym bardziej jest ono nieostre (nieprecyzyjne).)

Aproksymacje zbiorów - zastosowanie praktyczne

Dzielimy jabłka na dwa zbiory: X_d – zbiór jabłek dojrzałych i X_n – zbiór jabłek niedojrzałych. Cały czas pamiętamy, że robot rozróżnia tylko dwie cechy: kolor i wielkość.

	kolor	wielkość	dojrzałe
x_1	czerwone	duże	tak
x_2	żółte	średnie	tak
x_3	zielone	małe	nie
x_4	zielone	duże	tak
x_5	żółte	średnie	nie
x_6	czerwone	średnie	tak
x_7	żółte	duże	tak
x_8	czerwone	średnie	tak
x_9	żółte	małe	nie
x_{10}	żółte	małe	tak
x_{11}	czerwone	małe	tak
x_{12}	zielone	średnie	nie



Problem – jak nauczyć rozpoznawania, które jabłko jest dojrzałe (na podstawie koloru i wielkości), skoro w tabeli są jabłka dojrzałe i niedojrzałe przy tych samych cechach.

9.9. Aproksymacje:

Jeśli $SI = \{U, A, V, f\}$ jest systemem informacyjnym takim, że $B \subseteq A$ oraz $X \subseteq U$ to:

- B – dolnym przybliżeniem (aproksymacją) zbioru X w systemie informacyjnym SI nazywamy zbiór:

$$\underline{B}X = \{x \in U : I_{SI, B}(x) \subseteq X\}$$

Czyli będą to takie obiekty należące do $IND(B)$, które w całości zawierają się w zbiorze X .

O obiektach należących do dolnego przybliżenia mówimy, że NA PEWNO należą do danego pojęcia (danej klasy decyzyjnej).

- B – górnym przybliżeniem (aproksymacją) zbioru X w systemie informacyjnym SI nazywamy zbiór:

$$\overline{BX} = \{x \in U : I_{SI,B}(x) \cap X \neq \emptyset\}$$

Czyli będą to takie obiekty należące do $IND(B)$, które wystarczy, że mają część wspólną ze zbiorem X .

O obiektach należących do górnego przybliżenia mówimy, że **BYĆ MOŻE** należą do danego pojęcia (danej klasy decyzyjnej).

- B – pozytywnym obszarem (ang. positive area) zbioru X w systemie informacyjnym SI nazywamy zbiór:

$$POS_B(X) = \underline{BX}$$

- B – brzegiem (granicą) (ang. boundary) zbioru X w systemie informacyjnym SI nazywamy zbiór:

$$BN_B(X) = \overline{BX} - \underline{BX}$$

- B – negatywnym obszarem (ang. negative area) zbioru X w systemie informacyjnym SI nazywamy zbiór:

$$NEG_B X = U - \overline{BX}$$

Dolne przybliżenie pojęcia jest to więc pojęcie do którego należą wszystkie obiekty, co do których nie ma wątpliwości, że są one reprezentantami tego pojęcia w świetle posiadanej wiedzy. Do **górnego przybliżenia** należą obiekty, których nie można wykluczyć, że są reprezentantami tego pojęcia. **Brzegiem** zaś pojęcia są wszystkie te obiekty, co do których nie wiadomo czy są czy nie reprezentantami danego zbioru.

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
5	tak	nie	wysoka	nie
6	nie	tak	normalna	nie

$$IND(B) = \{\{1\}, \{2,5\}, \{3\}, \{4\}, \{6\}\}$$

$$X = X_{\text{tak}} + X_{\text{nie}}$$

$$\underline{BX} = \{Y \in IND(B) : Y \subseteq X\}$$

$$X_{\text{tak}} = \{1,2,3,4\}$$

$$X_{\text{nie}} = \{5,6\}$$

$$\underline{BX}_{\text{Tak}} = \{1,3,4\}$$

$$\underline{BX}_{\text{Nie}} = \{6\}$$

$$\overline{BX} = \{Y \in IND(B) : Y \cap X \neq \emptyset\}$$

$$X_{\text{tak}} = \{1,2,3,4\}$$

$$X_{\text{nie}} = \{5,6\}$$

$$\overline{BX}_{\text{Tak}} = \{1,2,3,4,5\}$$

$$\overline{BX}_{\text{Nie}} = \{2,5,6\}$$

Aproksymacje zbiorów - interpretacja na przykładzie

Niech:

$$B = \{ \text{kolor, wielkość} \}$$

$$X_n = \{ x_3, x_5, x_9, x_{12} \}$$

$$[x_1]_{IND(B)} = \{ x_1 \}$$

$$[x_2]_{IND(B)} = \{ x_2, x_5 \}$$

$$[x_3]_{IND(B)} = \{ x_3 \}$$

$$[x_4]_{IND(B)} = \{ x_4 \}$$

$$[x_5]_{IND(B)} = \{ x_2, x_5 \}$$

$$[x_6]_{IND(B)} = \{ x_6, x_8 \}$$

$$[x_7]_{IND(B)} = \{ x_7 \}$$

$$[x_8]_{IND(B)} = \{ x_6, x_8 \}$$

$$[x_9]_{IND(B)} = \{ x_9, x_{10} \}$$

$$[x_{10}]_{IND(B)} = \{ x_9, x_{10} \}$$

$$[x_{11}]_{IND(B)} = \{ x_{11} \}$$

$$[x_{12}]_{IND(B)} = \{ x_{12} \}$$

	kolor	wielkość	dojrzałe
x_1	czerwone	duże	tak
x_2	żółte	średnie	tak
x_3	zielone	małe	nie
x_4	zielone	duże	tak
x_5	żółte	średnie	nie
x_6	czerwone	średnie	tak
x_7	żółte	duże	tak
x_8	czerwone	średnie	tak
x_9	żółte	małe	nie
x_{10}	żółte	małe	tak
x_{11}	czerwone	małe	tak
x_{12}	zielone	średnie	nie

Przybliżenie dolne:

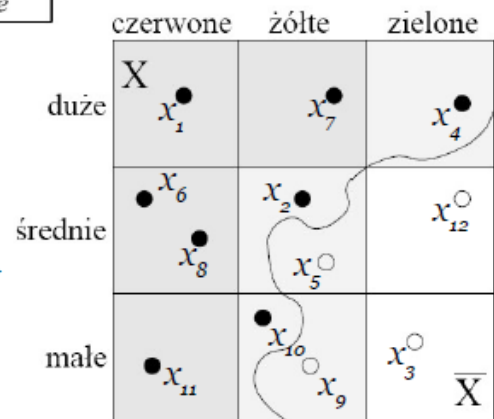
$$B_{IND(B)}X_n = \{ x_3, x_{12} \}$$

Przybliżenie górne:

$$B^{IND(B)}X_n = \{ x_2, x_3, x_5, x_9, x_{10}, x_{12} \}$$

Brzeg:

$$BN_N(X_n) = \{ x_2, x_5, x_9, x_{10} \}$$



9.10. Metody usuwania niespójności:

Wyróżnić można 5 metod usuwania niespójności w tablicach decyzyjnych:

1. Zwrócić się do EKSPERTA aby dla obiektów 2 i 5 podjął jedną decyzję.

Jest to sposób najprostszy przerzucający ciężar usunięcia niespójności z tablicy na eksperta. Niestety bardzo często zdarza się, że ekspert nie potrafi podjąć jednoznacznej decyzji. Twierdzi np. że dla takich atrybutów (parametrów) raz podejmuje decyzje 1 innym razem decyzje 2. W takim przypadku metoda ta nie daje rezultatu.

2. Utworzenie dwóch (lub więcej w przypadku ogólnym) spójnych tablic decyzyjnych, poprzez rozdzielenie sprzecznych obiektów.

Jest to jednak tylko pozorne rozwiązanie problemu. Powstaną dwa zbiory reguł dla pierwszej i drugiej tablicy. Reguły powstałe na podstawie obiektu 2 w tablicy pierwszej i reguła dla obiektu 5 w tablicy drugiej, będą sprzeczne.

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
6	nie	tak	normalna	nie

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
5	tak	nie	wysoka	nie
6	nie	tak	normalna	nie

3. Usunięcie obiektów będących przyczyną niespójności (metoda ilościowa).

Powstaje problem, który obiekt usunąć. Można posłużyć się tutaj metodą ilościową.

Metoda ilościowa

Wówczas usuniemy ten obiekt(-y), którego decyzja mniej razy była potwierdzana.

4. Można posłużyć się tutaj również metodą jakościową.

Usuniemy ten obiekt, którego wartość decyzja jest "mniej ważąca". "Mniej ważąca" to znaczy mająca mniejszą dokładność dolnego lub górnego przybliżenia.

Dla każdego $X \subseteq U$ i $B \subseteq A$ dokładność dolnego przybliżenia $\gamma_B(X)$ obliczymy ze wzoru:

$$\gamma_B(X) = \frac{|BX|}{|U|}$$

Dokładność górnego przybliżenia $\gamma^B(X)$ obliczymy ze wzoru:

$$\gamma^B(X) = \frac{|\bar{B}X|}{|U|}$$

Wówczas usuwamy ten obiekt, dla którego dokładności (górnego bądź dolnego) przybliżenia była mniejsza.

- | | |
|--|---|
| • $IND(B) = \{\{1\}\{2,5\}\{3\}\{4\}\{6\}\}$ | • $IND(B) = \{\{1\}\{2,\cancel{5}\}\{3\}\{4\}\{6\}\}$ |
| • $X_{tak} = \{1,2,3,4\}$ | • $X_{tak} = \{1,2,3,4\}$ |
| • $X_{nie} = \{5,6\}$ | • $X_{nie} = \{\cancel{5},6\}$ |
| • $\underline{BX}_{Tak} = \{1,3,4\}$ | • $\underline{BX}_{Tak} = \{1,3,4\}$ |
| • $\underline{BX}_{Nie} = \{6\}$ | • $\underline{BX}_{Nie} = \{6\}$ |
| • $\gamma_{Tak} = 3/6$ | • $\gamma_{Tak} = 3/6$ |
| • $\gamma_{Nie} = 1/6$ | • $\gamma_{Nie} = 1/6$ |

Usuniemy niespójność ze zbioru „NIE”

Spójna już teraz tablica decyzyjna wygląda następująco:

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
5	nie	tak	normalna	nie

5. Metoda tworzenia nowego atrybutu decyzyjnego (metoda uogólnionego atrybutu decyzyjnego)

Decyzja d wyznacza klasyfikację: $Class_A(d) = \{X_1, \dots, X_{r(d)}\}$, (gdzie (d) - to ilość różnych wartości atrybutu decyzyjnego.) Tworzymy nowy podział:

$$App - Class_A(d) = \{A|_{x_1}, \dots, A|_{x_{r(d)}}\} \cup \{Bd_A(\theta) : |\theta| > 1\}$$

Ten nowy podział tworzy tablice decyzyjną spójną.

Tabela nr 1. (niespójna) po dodaniu do systemu informacyjnego, nowego, uogólnionego atrybutu decyzyjnego wygląda następująco:

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak, nie
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
5	tak	nie	wysoka	tak, nie
6	nie	tak	normalna	nie

10. Macierz rozróżnialności:

Jeśli $SI = \{U, A, V, f\}$ jest systemem informacyjnym takim, że $U = \{u_1, u_2, \dots, u_n\}$ i $A = \{a_1, a_2, \dots, a_m\}$, to **macierz odróżnialności (rozróżnialności)** systemu informacyjnego SI $M(SI)$ (ang. discernibility matrix) definiujemy następująco:

$$M(SI) = (H_{i,j})_{i,j=1,\dots,n} = \{a \in A : f(u_i, a) \neq f(u_j, a)\}$$

dla $i, j = 1, \dots, n$, gdzie $n = |U|$.

Macierz odróżnialności jest dwuwymiarową macierzą kwadratową o wymiarach: $|U| \times |U|$.

Komórka $M(SI)[i, j]$ zawiera zbiór tych atrybutów, dla których obiekty uniwersum u_i i u_j mają różne wartości (są rozróżnialne przy pomocy tych atrybutów).

Dla KRS z przykładu 2 macierz $M(S) = [c_{ij}]$ jest następująca:

U	A			
	a	b	c	d
1	1	0	2	2
2	0	1	1	2
3	0	0	2	2
4	2	2	1	0

	1	2	3	4
1	\emptyset	a,b,c	a	a,b,c,d
2	a,b,c	\emptyset	b,c	a,b,d
3	a	b,c	\emptyset	a,b,c,d
4	a,b,c,d	a,b,d	a,b,c,d	\emptyset

Własności macierzy odróżnialności:

- macierz $M(SI)$ ma zawsze na przekątnej zbiory puste (\emptyset)
- macierz $M(SI)$ jest symetryczna względem przekątnej
- każdy element macierzy $M(SI)$ jest zbiorem
- rozmiar macierzy rośnie w sposób kwadratowy wraz ze wzrostem liczby obiektów w systemie informacyjnym

Generowanie macierzy odróżnialności

Wejście: $A = (U, A)$ system informacyjny taki, że $U = \{u_1, \dots, u_n\}$ i $A = \{a_1, \dots, a_m\}$.

Wyjście: $M(A) = (C_{ij})_{i,j=1,\dots,n}$ macierz odróżnialności systemu A , przy czym $M(A)$ ma obliczone tylko te pola C_{ij} dla których $1 \leq j < i \leq n$.

Metoda:

For $i=1$ to n do

For $j=1$ to $i-1$ do

Wstaw do C_{ij} atrybutv. na których różnią się obiektv u_j i u_i .

11. Wyznaczanie jądra i reduktów:

Kiedy atrybut jest niezbędny?

Niech $B \subseteq A$ i $a \in B$. Mówimy, że atrybut a jest zbędny w B , gdy $IND(B) = IND(B - \{a\})$.
W przeciwnym przypadku atrybut a jest niezbędnym w B .

Zbiór atrybutów B jest *niezależny*, gdy dla każdego $a \in B$ atrybut a jest niezbędnym.
W przeciwnym przypadku zbiór jest *zależny*.

Definicja jądra

Niech dany będzie system informacyjny $SI = (U, A)$ oraz zbiór atrybutów $B \subseteq A$. Zbiór wszystkich niezbędnych atrybutów w B nazywamy *jądrem* (rdzeniem) i oznaczamy przez $CORE(B)$.

Definicja reduktu

Podzbiór atrybutów $B \subseteq A$ nazywamy *reduktem* zbioru atrybutów A , gdy zbiór atrybutów B jest *niezależny* oraz $IND(B) = IND(A)$.

Zbiór wszystkich reduktów oznaczamy przez $RED(A)$.

Redukt to najmniejszy zbiór atrybutów, przy którym zostaje zachowana dotychczasowa klasyfikacja (rozróżnialność) obiektów

Ważne!

Redukt musi spełniać *dwa* kryteria:

1. musi być *niezależnym zbiorem atrybutów* (tylko atrybuty niezbędne),
2. musi *zachowywać taką samą rozróżnialność obiektów* jak zbiór redukowany.

Związek pomiędzy jądrem a reduktem

Jądro systemu informacyjnego rozpatrywanego dla podzbioru atrybutów $B \subseteq A$ jest częścią wspólną wszystkich reduktów tego systemu.

$$CORE(B) = \bigcap RED(A).$$

Uwaga! To właściwość wiążąca jądro i redukty a nie definicja jądra!

Przykład

Pacjent	Ból głowy (g)	Ból mięśni (m)	Temperatura (t)	Grypa (c)
1	nie	tak	wysoka	tak
2	tak	nie	wysoka	tak
3	tak	tak	bardzo wysoka	tak
4	nie	tak	bardzo wysoka	tak
6	nie	tak	normalna	nie

Zbiór wszystkich reduktów zbioru atrybutów $\{g, m, t, c\}$ systemu informacyjnego z tabeli 1 wynosi: $RED_{SI}(\{g, m, t, c\}) = \{g, t, c\}$.

Aby udowodnić, że zbiór $\{g, t, c\}$ jest reduktem należy pokazać, że zachodzą warunki z definicji:

- $IND_{SI}(\{g, m, t, c\}) = IND_{SI}(\{g, t, c\})$,

Możemy to pokazać, usuwając z tego zbioru kolejne atrybuty i sprawdzając czy relacja nierozróżnialności względem takiego okrojonego zbioru jest różna od relacji nierozróżnialności względem całego zbioru atrybutów. Jeżeli tak będzie, to zbiór $\{g, t, c\}$ będzie reduktem.

Wyznaczanie jądra z definicji

U	A			
	a	b	c	d
1	1	0	2	2
2	0	1	1	2
3	0	0	2	2
4	2	2	1	0

$$U/IND(B) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

$$U/IND(B-\{a\}) = \{\{1, 3\}, \{2\}, \{4\}\} \rightarrow a \text{ jest niezbędny}$$

$$U/IND(B-\{b\}) = \{\{1\}, \{2\}, \{3\}, \{4\}\} \rightarrow b \text{ jest zbędny}$$

$$U/IND(B-\{c\}) = \{\{1\}, \{2\}, \{3\}, \{4\}\} \rightarrow c \text{ jest zbędny}$$

Zakładamy, że $B = \{a, b, c\}$

$$\text{Jądro } CORE(B) = \{a\}$$

Ogólny algorytm wyznaczania jądra z definicji

1. Wyznacz klasy abstrakcji relacji nierozróżnialności $U/IND(B)$, gdzie B jest to zbiór wszystkich rozważanych atrybutów.
2. Wyznacz klasy abstrakcji z pominięciem i -tego atrybutu $U/IND(B-\{a_i\})$.
3. Jeżeli $U/IND(B) = U/IND(B-\{a_i\})$ to
4. Atrybut a_i jest zbędny,
5. W przeciwnym wypadku
6. Atrybut a_i jest niezbędny i wchodzi do jądra $CORE(B)$.
7. Powtarzaj pkt. 2, aż wykorzystane zostaną wszystkie atrybuty z B .

Wyznaczanie reduktów z definicji

U	A			
	a	b	c	d
1	1	0	2	2
2	0	1	1	2
3	0	0	2	2
4	2	2	1	0

Zakładamy, że $B = \{a, b, c\}$

Potencjalne redukty to:

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

Ale wiemy już, że $CORE(B) = \{a\}$

Skoro jądro utrzymuje rozróżnialność obiektów w systemie, to nie możemy atrybutów z jądra zredukować.

Pozwala to zawęzić zbiór potencjalnych reduktów do:

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

Samo jądro jest interesującym kandydatem na redukt. Z definicji spełnia już *pierwszy* warunek dla reduktu – jest zbioremnie zależnym.

Czy jednak $U/IND(\{a\}) = U/IND(\{a, b, c\})$? Sprawdźmy *drugi* warunek:

$$U/IND(\{a, b, c\}) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

$$U/IND(\{a\}) = \{\{1\}, \{2, 3\}, \{4\}\} \rightarrow \text{Jądro } CORE(B) = \{a\} \text{ nie jest reduktem}$$

Czy $B_1 = \{a, b\}$ jest reduktem?

1. Czy B_1 jest niezależny?

$$U/IND(B_1) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

$$U/IND(B_1-\{a\}) = \{\{1, 3\}, \{2\}, \{4\}\} \rightarrow \{a\} \text{ jest niezbędny}$$

$$U/IND(B_1-\{b\}) = \{\{1\}, \{2, 3\}, \{4\}\} \rightarrow \{b\} \text{ jest niezbędny}$$

Zbiór B_1 jest niezależny, spełnia pierwszy warunek reduktu

2. Czy B_1 zachowuje taką samą rozróżnialność obiektów jak B ?

$$U/IND(\{a, b, c\}) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

$$U/IND(B_1) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

Widzimy, że:

$$U/IND(\{a, b, c\}) = U/IND(B_1)$$

Zatem

Zbiór B_1 jest reduktem

Czy $B_2 = \{a, c\}$ jest reduktem?

1. Czy B_2 jest niezależny?

$$U/IND(B_2) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

$$U/IND(B_2 - \{a\}) = \{\{1, 3\}, \{2, 4\}\} \rightarrow \{a\} \text{ jest niezbędny}$$

$$U/IND(B_2 - \{c\}) = \{\{1\}, \{2, 3\}, \{4\}\} \rightarrow \{c\} \text{ jest niezbędny}$$

Zbiór B_2 jest niezależny, spełnia pierwszy warunek reduktu

2. Czy B_2 zachowuje taką samą rozróżnialność obiektów jak B ?

$$U/IND(\{a, b, c\}) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

$$U/IND(B_2) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

Widzimy, że:

$$U/IND(\{a, b, c\}) = U/IND(B_2)$$

Zatem

Zbiór B_2 jest reduktem

Ostatecznie, dla rozważanego systemu informacyjnego i zbioru $B = \{a, b, c\}$ zbiór reduktów $RED(B)$:

$$RED(B) = \{\{a, b\}, \{a, c\}\}$$

Ogólny algorytm wyznaczania reduktów z definicji

1. Wyznacz klasy abstrakcji $U/IND(B)$, gdzie B jest to zbiór wszystkich rozważanych atrybutów.
2. Sprawdź, czy jądro $CORE(B)$ nie jest reduktem:
3. Ponieważ jądro to zbiór atrybutów niezbędnych, to sprawdź, czy $U/IND(B) = U/IND(CORE(B))$, jeżeli tak to jądro to jedyny redukt i przejdź do Punktu 6.
4. Sprawdź kolejne podzbiory atrybutów $B_i \subseteq B$.
5. Jeżeli podzbiór B_i jest niezależny to:
Jeżeli $U/IND(B) = U/IND(B_i)$ to:
Dopisz podzbiór B_i do zbioru reduktów.
6. Koniec.

Macierz nierozróżnialności - przykład

Zakładamy, że $B = \{a, b, c\}$

$$SI = (U, A)$$

U	A			
	a	b	c	d
1	1	0	2	2
2	0	1	1	2
3	0	0	2	2
4	2	2	1	0

$$M(SI) = [c_{ij}]_{n \times n}$$

	1	2	3	4
1	\emptyset	a,b,c	a	a,b,c,d
2	a,b,c	\emptyset	b,c	a,b,d
3	a	b,c	\emptyset	a,b,c,d
4	a,b,c,d	a,b,d	a,b,c,d	\emptyset

Wyznaczenie jądra i reduktów z macierzy nierozróżnialności

► Wyznaczanie jądra $CORE(B)$:

Do rdzenia wchodzi atrybuty występujące w macierzy nierozróżnialności pojedynczo.

$$CORE(B) = \{ a \in A : c_{ij} = \{ a \}, \text{ dla pewnego } 1 \leq i, j \leq n \}$$

► Wyznaczanie reduktów $RED(B)$:

Pewien podzbiór atrybutów $C \subseteq B$ jest reduktem jeśli jest minimalny (w sensie zawierania zbiorów) oraz posiada niepuste przecięcie z każdym niepustym elementem macierzy $M(SI)$.

Macierz nierozróżnialności - przykład

$M(SI) = [c_{ij}]_{n \times n}$

	1	2	3	4
1	\emptyset	a,b,c	a	a,b,c,d
2	a,b,c	\emptyset	b,c	a,b,d
3	a	b,c	\emptyset	a,b,c,d
4	a,b,c,d	a,b,d	a,b,c,d	\emptyset

$$CORE(B) = \{ a \}, \text{ bo } c_{13} = \{ a \}$$

$$RED(B) = \{ \{ a, b \}, \{ a, c \} \}$$

ponieważ:

$$\{ a, b \} \cap c_{ij} \neq \emptyset \text{ i } \{ a, c \} \cap c_{ij} \neq \emptyset$$

i te zbiory są minimalne.

12. Wyznaczanie reguł minimalnych:

Tablica decyzyjna

Szczególnym rodzajem systemów informacyjnych są tablice decyzyjne (TD). Tablicą decyzyjną nazywamy uporządkowaną piątkę:

$$TD = (U, C, D, V, f)$$

gdzie:

- $C, D \subset A; C \neq \emptyset; C \cup D = A; C \cap D = \emptyset$,
- elementy zbioru C nazywamy atrybutami warunkowymi,
- elementy zbioru D nazywamy atrybutami decyzyjnymi,
- f nazywamy funkcją decyzyjną.
- interpretacja U oraz V jest taka sama jak w przypadku systemu informacyjnego, ponadto poszczególne wartości v dziedzin atrybutów $D (v \in V_D)$ będziemy nazywać **klasami decyzyjnymi**.

5.1 Tablice decyzyjne deterministyczne i nondeterministyczne

Każdy obiekt $u \in U$ tablicy decyzyjnej $TD = (U, C, D, V, f)$ może zostać zapisany w postaci zdania warunkowego (postaci: *jeżeli warunki to decyzja*) i być traktowany jako **reguła decyzyjna**.

Regułą decyzyjną w tablicy decyzyjnej TD nazywamy funkcję: $g : C \cup D \rightarrow V$ jeżeli istnieje $x \in U$, taki, że $g = f_x$.
Obcięcie g do C ($g|C$) oraz g do D ($g|D$) nazywamy odpowiednio warunkami oraz decyzjami reguły decyzyjnej g .

Przykład

Z przykładowej tablicy decyzyjnej z tabeli 1 możemy wyprowadzić następujące reguły (odpowiadające konkretnym obiektom):

1. jeżeli (g ="nie") i (m ="tak") i (t ="wysoka") to (c ="tak")
2. jeżeli (g ="tak") i (m ="nie") i (t ="wysoka") to (c ="tak")
3. jeżeli (g ="tak") i (m ="tak") i (t ="bardzo wysoka") to (c ="tak")
4. jeżeli (g ="nie") i (m ="tak") i (t ="bardzo wysoka") to (c ="tak")
5. jeżeli (g ="tak") i (m ="nie") i (t ="wysoka") to (c ="nie")
6. jeżeli (g ="nie") i (m ="tak") i (t ="normalna") to (c ="nie")

Reguły decyzyjne można dzielić na wiele różnych grup biorąc pod uwagę różne kryteria. Jeden z podziałów wyróżnia dwie grupy reguł:

- **reguły deterministyczne**

Reguła w tablicy decyzyjnej TD jest *deterministyczna*, gdy równość atrybutów warunkowych implikuje równość atrybutów decyzyjnych. Fakt ten możemy wyrazić przy pomocy następującej zależności dla obiektów tablicy decyzyjnej:

$$\forall_{x,y \in U} (\forall_{c \in C} (f(x,c) = f(y,c)) \Rightarrow \forall_{d \in D} (f(x,d) = f(y,d)))$$

- **reguły niedeterministyczne**

Reguła w tablicy decyzyjnej TD jest *niedeterministyczna*, gdy równość atrybutów warunkowych nie implikuje równości atrybutów decyzyjnych, co można wyrazić następującą zależnością dla obiektów tablicy decyzyjnej:

$$\forall_{x,y \in U} (\forall_{c \in C} (f(x,c) = f(y,c)) \wedge \exists_{d \in D} (f(x,d) \neq f(y,d)))$$

Tablica decyzyjna jest deterministyczna (dobrze określona, spójna), gdy wszystkie reguły w niej zawarte są deterministyczne, w przeciwnym przypadku jest niedeterministyczna ("źle określona, niespójna).

Reguły minimalne

Znanym algorytmem generującym reguły na podstawie tablicy decyzyjnej jest algorytm zamieszczony w pracy Z. Pawlaka i A. Skowrona z 1993 roku.

Ponieważ algorytm działa dla tablic spójnych, na początku sprawdza się spójność tablicy decyzyjnej. W przypadku wystąpienia niespójności usuwa się ją za pomocą **uogólnionego atrybutu decyzyjnego** lub **metody jakościowej** usuwania niespójności z tablicy decyzyjnej.

student	a	b	c	d
x ₁	a ₁	b ₁	c ₁	T
x ₂	a ₁	b ₁	c ₂	T
x ₃	a ₂	b ₁	c ₃	T
x ₄	a ₂	b ₁	c ₄	N
x ₅	a ₁	b ₂	c ₁	N
x ₆	a ₁	b ₂	c ₂	T
x ₇	a ₂	b ₂	c ₃	T
x ₈	a ₂	b ₂	c ₄	N

- r1: $a = a_1 \wedge b = b_1 \rightarrow d = T$
- r2: $b = b_1 \wedge c = c_1 \rightarrow d = T$
- r3: $b = b_1 \wedge c = c_2 \rightarrow d = T$
- r4: $c = c_3 \rightarrow d = T$
- r5: $c = c_4 \rightarrow d = N$
- r6: $b = b_2 \wedge c = c_1 \rightarrow d = N$
- r7: $c = c_2 \rightarrow d = T$

Generowanie reguł minimalnych:

1. Doprowadź tablicę do spójności, np. wprowadzając uogólniony atrybut decyzyjny.
2. Usuń identyczne obiekty.
3. Utwórz macierz nierozróżnialności
4. Dla każdej wartości atrybutu decyzyjnego:
5. Utwórz uogólnioną macierz nierozróżnialności względem decyzji.
6. Zapisz funkcję nierozróżnialności, zminimalizuj ją.
7. Zapisz regułę decyzyjną na podstawie zminimalizowanej funkcji rozróżnialności.

Algebra Boola jest to struktura matematyczna $A = (X, +, *, \neg, 0, 1)$ spełniająca następujące aksjomaty:

- łączność i przemienność:
 - $x + y = y + x$
 - $x * y = y * x$
 - $(x + y) + z = x + (y + z)$
 - $(x * y) * z = x * (y * z)$
- 0 jest elementem neutralnym dla + : $x + 0 = x$
- 1 jest elementem neutralnym dla * : $x * 1 = x$
- $x + (\neg x) = 1$
- $x * (\neg x) = 0$
- rozdzielność + i * względem siebie:
 - $x * (y + z) = (x * y) + (x * z)$
 - $x + (y * z) = (x + y) * (x + z)$
- dwa działania \neg się znoszą: $\neg\neg x = x$
- prawa de Morgana
 - $\neg(x * y) = (\neg x) + (\neg y)$
 - $\neg(x + y) = (\neg x) * (\neg y)$

13. Rachunek predykatów:

13.1. Rachunek zdań pierwszego rzędu

Rachunek zdań jest także jednym ze sposobów zapisu wiedzy. Można by stwierdzić, że jest on systemem wyrażen będących formułami prawdziwymi, w którym nie stosuje się konkretnych zdań, lecz posługuje się tzw. **zmiennymi zdaniowymi reprezentującymi zdania**.

Cała teoria opiera się na klasycznej logice dwuwartościowej, zgodnie z którą, za zmienne zdaniowe można podstawiać takie zdania, którym odpowiada wartość logiczna **TRUE (prawda)** lub **FALSE (fałsz)**, tzn. takie, które uznane są odpowiednio za prawdziwe lub fałszywe.

Funkcją zdaniową [lub **predykatem**] jest wyrażenie zawierające zmienne, z którego otrzymamy zdania po podstawieniu za zmienne odpowiednich stałych.

Mówimy, że **przedmiot spełnia** daną funkcję zdaniową wtedy i tylko wtedy, gdy z funkcji tej po podstawieniu nazwy tego przedmiotu otrzymujemy zdanie prawdziwe.

RACHUNEK PREDYKATÓW = JĘZYK PREDYKATÓW:

JĘZYK PREDYKATÓW

$J_p = \langle A, G \rangle$

Alfabet A:

- stałe indywidualne (np. Kraków, Kowalski)
- zmienne indywidualne x_1, \dots, x_n (np. miasto, osoba)
- n-argumentowe symbole funkcyjne f_1, \dots, f_n (np. matka(Ewa))
- n-argumentowe symbole predykatów p_1, \dots, p_n (np. jest ojcem(Adam, Piotr))
- $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- symbole pomocnicze – przecinek, nawiasy

Gramatyka G:

(1) Syntaktyka – sposób tworzenia zdań i słów.

Słowa języka predykatów są termy – Tm. Do zbioru termów należą:

- stałe indywidualne
- zmienne indywidualne
- funkcje określone na zbiorze termów $f(t_1, \dots, t_n)$

Zdania w języku predykatów noszą nazwę formuł.

Zmienna występuje w dwóch postaciach:

- związanej – jeśli jest objęta działaniem kwantyfikatora
- wolnej – jeśli przynajmniej jedno jej wystąpienie jest niezwiązane.

Formuła, w której nie występują zmienne wolne nosi nazwę zdania.

np. $\exists x \forall y \exists z (x + y = z)$

Predykat określony na zbiorze termów to formuła atomowa. Z formuł atomowych tworzone są formuły złożone z uwzględnieniem w alfabecie funkcyjnych.

(2) Semantyka – nadaje znaczenie słowom i zdaniom języka, jest interpretacją wyrażen języka.

Interpretacja języka predykatów:

$\langle D, m \rangle$

D – dziedzina

m – funkcja przyporządkowująca każdej stałej indywidualnej element z dziedziny D

m:

Każdemu symbolowi funkcyjnemu przyporządkowuje odwzorowanie do dziedziny D.

Każdemu symbolowi predykatów przyporządkowuje odwzorowanie do zbioru $\{0, 1\}$.

Każdemu symbolowi stałej zdaniowej przyporządkowuje wartość prawda lub fałsz.

Przez $As(I)$ będziemy oznaczać zbiór wartościowań I przy interpretacji $I = \langle D, m \rangle$. Wartość funkcji V wszystkich występowania zmiennych będą należeć do zbioru interpretacji $V \in As(I)$, $V(x_k) \in D$.

14. Czym jest predykat? Co może być argumentem predykatu?

PREDYKAT - nazwa reprezentująca jakąś własność lub regułę.

np: Posiada kota (Mercedes) = Mercedes posiada kota

Jest siostrą (Ewa, Magola) = Ewa jest siostrą Magoly

Istotne dla predykatów jest występowanie w nich kwantyfikatorów: \forall (dla każdego), \exists (istnieje)

Argumentami predykatów mogą być stałe indywidualne (nazwy, obiekty), zmienne indywidualne (dłuszone symbole) oraz predykaty. Złożone wyrażenie rachunków predykatów noszą nazwę formuł.

Wyróżnia się rachunek predykatów I-go i II-go rzędu. Rachunek predykatów I-go rzędu operuje na pojęciach abstrakcyjnych, posiada mechanizmy pozwalające opisać prawa, którym podlegają obiekty systemu. Funkcje zdaniowe reprezentowane są za pomocą reguł zawierających implikację. Np.:

$(p \rightarrow q)$

gdzie p i q to predykaty, to reguła postaci:

Jeżeli p To q .

- Niezawodny schemat rozumowy - to taki schemat, w którym prawdziwość przesłania gwarantuje prawdziwość wniosku. Nie możliwe jest by otrzymać prawdziwy wniosek przy fałszywej przesłance. $\neg\neg A \equiv A$.
- Funktor - wyrażenie, które posiada argumenty i które tworzy razem z tymi argumentami nowe zdanie lub inny funktor. Przyjmuje się, że samodzielnymi kategoriami syntaktycznymi są nazwy i zdania a niesamodzielnymi funktry tzn. ich znaczenie zależy od argumentów.

13.2. Metody dowodzenia prawdziwości schematów

Schemat wnioskowania może być:

- ♦ **Formalny** - czyli zawierający wyrażenia zbudowane wyłącznie ze stałych logicznych i zmiennych,
- ♦ **Niezawodny** - prowadzący zawsze od prawdziwych przesłanek do prawdziwych wniosków,
- ♦ **Logiczny** - gdy jest poprawny i niezawodny.

② Metody badania niezawodności schematów wnioskowania:

- Metoda zerojedynkowa,
- Skrócona metoda zerojedynkowa.

③ Metody dowodu niezawodności schematów - metoda założeniowa:

- Założeniowa metoda dowodu "wprost",
- Założeniowa metoda dowodu "nie wprost".

13.2.1 Metoda zerojedynkowa

Korzystamy w niej ze związków zachodzących między wartościami logicznymi zdań wchodzących w skład wyrażenia.

p	q	$\neg p$	$\neg q$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \Leftrightarrow q$
0	0	1	1	0	0	1	1
0	1	1	0	1	0	1	0
1	0	0	1	1	0	0	0
1	1	0	0	1	1	1	1

Aby rozstrzygnąć, czy dany schemat jest tautologią, należy rozważyć wszystkie możliwe kombinacje wartości logicznych zmiennych w niej występujących. Jeżeli w każdym przypadku wartość formuły (wyrażenia logiczne połączone funktorami) wynosi 1, to ta formuła jest tautologią.

PRZYKŁAD : Sprawdzić prawdziwość następującego schematu:

$$(\neg p \vee q) \Leftrightarrow (p \rightarrow q)$$

p	q	$\neg p$	$\neg p \vee q$	$p \rightarrow q$	$(\neg p \vee q) \Leftrightarrow (p \rightarrow q)$
1	1	0	1	1	1
1	0	0	0	0	1
0	1	1	1	1	1
0	0	1	1	1	1

Wartości logiczne w ostatniej kolumnie są identyczne, co dowodzi, że schemat jest prawdziwy dla dowolnych wartości logicznych wyrażeń wchodzących w skład schematu. czyli jest tautologią.

PRZYKŁAD :

Sprawdzić następujący schemat logiczny:


$$\neg(p \wedge q) \rightarrow (\neg p \wedge \neg q)$$

p	q	$\neg p$	$\neg q$	$\neg(p \wedge q)$	$\neg p \wedge \neg q$	$\neg(p \wedge q) \rightarrow (\neg p \wedge \neg q)$
1	1	0	0	0	0	1
1	0	0	1	1	0	0
0	1	1	0	1	0	0
0	0	1	1	1	1	1

Ostatnia kolumna przyjmuje różne wartości w zależności od wartości jakie przyjmują zmienne wchodzące w jej skład, czyli schemat nie jest tautologią.


Skróconym sprawdzaniem zerjedynkowym wykazujemy, że wyrażenie rachunku zdań o postaci implikacji jest prawem logicznym, gdy jest wykluczone, by dla jakiegoś układu wartości logicznych przyporządkowanego zmiennym poprzednik tej implikacji był prawdziwy a jej następnik fałszywy. Zakładając np. prawdziwość poprzednika takiej implikacji i wykazując, że wtedy jej następnik musi być prawdziwy, wykazujemy tym samym, że implikacja ta jest prawem logicznym. Podobnie zakładając fałszywość następnika danej implikacji i wykazując, że jej poprzednik musi być wtedy fałszywy, wykazujemy tym samym, że jest wykluczone, by dla jakiegoś układu wartości logicznych przyporządkowanego jej zmiennym jej poprzednik był prawdziwy, a jej następnik fałszywy, a więc wykazujemy, że jest ona prawem logicznym. Oto przykłady takiego sprawdzania.

$$\neg(p \vee q) \rightarrow (\neg p \vee \neg q)$$

$$1 \ 0 \ 0 \ 0 \quad 1 \ 0 \ 1 \ 1 \ 0$$


Zakładamy, że poprzednik tej implikacji, tj. wyrażenie " $\neg(p \vee q)$ ", ma wartość 1. Wtedy wyrażenie " $p \vee q$ " ma wartość 0, a wówczas zarówno " p " jak i " q " mają wartość 0. Wtedy zaś zarówno " $\neg p$ " jak i " $\neg q$ " mają wartość 1. Jeśli poprzednik tej implikacji jest prawdziwy, to i jej następnik jest prawdziwy, a więc ta implikacja jest prawem logicznym. Strzałka umieszczona pod sprawdzeniem wskazuje na jego kierunek: zakładając prawdziwość poprzednika dochodzimy do stwierdzenia, że następnik musi być wtedy prawdziwy.

$$(\neg p \wedge \neg q) \rightarrow \neg(p \wedge q)$$

$$0 \ 1 \ 0 \ 0 \ 1 \quad 0 \ 1 \ 1 \ 1$$


Zakładamy, że następnik implikacji, tj. wyrażenie " $\neg(p \wedge q)$ " ma wartość 0. Wtedy koniunkcja " $p \wedge q$ " ma wartość 1, a więc oba jej człony mają też wartość 1. Negacje tych członów, tj. wyrażenia " $\neg p$ " i " $\neg q$ " mają wtedy wartość 0, a więc i ich koniunkcja " $\neg p \wedge \neg q$ ", stanowiąca poprzednik implikacji, ma wartość 0. Jeśli następnik tej implikacji jest fałszywy, to i jej poprzednik jest fałszywy, a więc implikacja ta jest prawem logicznym. Strzałka umieszczona pod sprawdzeniem wskazuje na jego kierunek: zakładając fałszywość następnika dochodzimy do stwierdzenia, że poprzednik musi być wtedy fałszywy.

W metodzie dowodów założeniowych dodatkowo przyjmujemy, że niektóre schematy logiczne (wyszczególnione poniżej) są niezawodne (nazywamy je **schematami pierwotnymi**, zdania stwierdzające zaś, że są one schematami niezawodnymi nazywamy **regułami pierwotnymi**). Są one dobrane w taki sposób by ich niezawodność nie budziła żadnych wątpliwości.

1. Regułą odrywania (RO) opisana powyżej.

2. Reguła dołączania koniunkcji (DK) $(p) \wedge (q) \rightarrow (p \wedge q)$

$$(p \wedge q) \rightarrow p$$

3. Reguła opuszczania koniunkcji (OK) $(p \wedge q) \rightarrow q$

$$p \rightarrow (p \vee q)$$

4. Reguła dołączania alternatywy (DA) $q \rightarrow (p \vee q)$

$$((p \vee q) \wedge \neg p) \rightarrow q$$

5. Reguła opuszczania alternatywy (OA) $((p \vee q) \wedge \neg q) \rightarrow p$

6. Reguła dołączania równoważności (DE) $((p \rightarrow q) \wedge (q \rightarrow p)) \rightarrow (p \leftrightarrow q)$

$$(p \leftrightarrow q) \rightarrow (p \rightarrow q)$$

7. Reguła opuszczania równoważności (OE) $(p \leftrightarrow q) \rightarrow (q \rightarrow p)$

Wyróżniamy dwie techniki metody założeniowej:

- założeniowy dowód "nie wprost"
- założeniowy dowód "wprost"

METODA ZAŁOŻENIOWEGO DOWODU "NIE WPROST"

polega na tym, że z twierdzenia W w postaci $w_1 \rightarrow (w_2 \rightarrow w_3 \rightarrow \dots (w_n \rightarrow W))$ wypisujemy najpierw wyrażenia w_1, \dots, w_n i następnie negację wyrażenia W . Dalsze wyrażenia dołączamy do dowodu korzystając z przyjętych reguł i twierdzeń wcześniej udowodnionych. Dowód jest zakończony jeżeli wystąpią w nim dwa wyrażenia, z których jedno jest negacją drugiego.

Dowód niezawodności schematu:

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

zapisujemy w następujący sposób:

1	$p \rightarrow q$	
2	$q \rightarrow r$	
3	$\neg(p \rightarrow r)$	DN
4	$\neg p \vee q$	[z reguły zastępowania implikacji ZI(1)]
5	$\neg q \vee r$	[z reguły zastępowania implikacji ZI(2)]
6	$\neg(\neg p \vee r)$	[z reguły zastępowania implikacji ZI(3)]
7	$\neg\neg p \wedge \neg r$	[z reguły negowania alternatywy NA(6)]
8	$p \wedge \neg r$	[z prawa podwójnej negacji PN(7)]
9	p	[z prawa odrywania koniunkcji OK(8)]
10	$\neg r$	[z prawa odrywania koniunkcji OK(8)]
11	q	[z prawa odrywania alternatywy OA(4,9)]
12	$\neg q$	[z prawa odrywania alternatywy OA(5,10)]

Zasza sprzeczność dla zdań 11 oraz 12, a więc dowód był prawdziwy, a jedynie negacji tezy doprowadziła do sprzeczności. Skoro więc zaprzeczona teza jest niemożliwa, to prawdziwa jest niezaprzeczona teza.

METODA ZAŁOŻENIOWEGO DOWODU "WPROST"

polega na tym, że z twierdzenia W w postaci $w_1 \rightarrow (w_2 \rightarrow w_3 \rightarrow \dots (w_n \rightarrow W))$ wypisujemy najpierw wyrażenia w_1, \dots, w_n potem zaś wyrażenia, na dołączenie których pozwalają przyjęte reguły. Wolno też dołączyć do dowodu twierdzenia wcześniej udowodnione. Dowód jest zakończony, gdy wystąpi w nim wyrażenie W .

Można inaczej powiedzieć, że w metodzie założeniowej (wprost) rozpatrywany schemat uznajemy za niezawodny, gdy w wyniku kolejnych działań, podczas których uzyskujemy schematy już udowodnione jako niezawodne, ostatecznie uzyskamy cel wniosowania (konkluzję całego wyrażenia). Nie można jednak na niej polegać w przypadku wykazywania zawodności schematów.

Dowód niezawodności schematu:

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

zapisujemy w następujący sposób:

1	$p \rightarrow q$	
2	$q \rightarrow r$	
3	p	
4	q	[z reguły odrywania RO(1, 3)]
5	r	[z reguły odrywania RO(2, 4)]

Zgodnie z metodą założeniową dowód rozpoczynać powinno wypisanie założeń, którymi są przesłanki dowodzonego schematu i poprzednik jej wniosku. Komentarze umieszczone na boku mówią o tym, z których wierszy wcześniejszych kolejne wiersze są otrzymane przy pomocy reguły odrywania. Dowód jest zakończony, gdyż otrzymaliśmy następnik wniosku.

13.3 Rachunek predykatów II rzędu

Kwantyfikatory są to najzwyczajniejsze w świecie stale (oczywiście logiczne), występujące sobie w (noszącym znamiona graficznego rozpisu sensu zdania) rachunku kwantyfikatorów, a oznaczane przez więcej niż wielu wytrawnych Logików w następujący sposób:

- **kwantyfikator ogólny**, zapisywany jako \forall , czytany jako: "dla każdego..."
- **kwantyfikator szczegółowy** (egzystencjalny), zapisywany jako \exists , czytany jako: "istnieje taki ..., że..."

NAZWY- są dowolne zmienne - pojedyncze rzeczy, występujące w zdaniu i oznaczamy je małymi literami w następujący sposób: " x, y, z, \dots "

PREDYKATY - są to zmienne - własności NAZW i relacje między tymi NAZWAMI zachodzące. Oznaczamy je wielkimi literami: " P, Q, R, S, \dots "

Predykaty reprezentują w wyrażeniu rachunku kwantyfikatorów albo NAZWE (zapisuje się to zawsze tak: $P(x)$), albo też relacje pomiędzy NAZWAMI (zapis: $P(x, y)$).

SCHEMAT ZDANIOWY - jest to symboliczny zapis odzwierciedlający zawartość zdania, np.:

- $\forall x P(x)$ - czytaj jako: "Dla każdego x , x jest Ptakiem"
- $\exists y Q(y)$ - czytaj jako: "Istnieje taki y , że y jest Kurą"

Wypisujemy sobie zmienne nazwowe (NAZWY), którymi są zawsze tylko te wszystkie podmioty (rzeczowniki), w stosunku do których inne części zdania (mogą nimi być także rzeczowniki w formie dopełnienia), pełnią funkcje opisowa:

x - Kubuś

y - Antykubuś

z - czas

❶ Tworzenie zmiennych predykatowych (PREDYKATÓW):

$K(x)$ - x jest Kubusiem

$A(y)$ - y jest Antykubusiem

$C(z)$ - z jest czasem

$W(x, y)$ - x widział y

$G(y, z)$ - y gonił z

❷ Przekształćmy sobie nasze zdanie tak, aby przybrało formę ułatwiającą nam dopasowanie odpowiednich kwantyfikatorów :

”(Jeden) Kubuś widział (jednego) Antykubusia, goniącego (jeden) czas.”

Mamy teraz pewność, że:

- Kubuś jest jeden, więc możemy powiedzieć: *”Istnieje taki x , że x jest Kubusiem”* i zapisać to zaraz w schemacie, używając w tym celu MAŁEGO kwantyfikatora.
- Antykubuś jest jeden, więc możemy powiedzieć: *”Istnieje taki y , że y jest Antykubusiem”* i zapisać to zaraz w schemacie, używając w tym celu MAŁEGO kwantyfikatora.
- czas jest jeden, więc możemy powiedzieć: *”Istnieje taki z , że z jest czasem”* i zapisać to zaraz w schemacie, używając w tym celu MAŁEGO kwantyfikatora.

Przystępujemy do zapisania naszego zdania w postaci schematu kwantyfikatorowego:

$$\exists x \{K(x) \wedge \exists y [A(y) \wedge W(x, y) \wedge \exists z (C(z) \wedge G(y, z))]\}$$

gdzie:

$K(x)$ - x jest Kubusiem

$A(y)$ - y jest Antykubusiem

$C(z)$ - z jest czasem

$W(x, y)$ - x widział y

$G(y, z)$ - y gonił z

ale po kolei...

$$\exists x K(x)$$

czytaj jako: *”Istnieje taki x , że x jest Kubusiem...”*

$$\exists x \{K(x) \wedge \exists y [A(y) \dots$$

czytaj jako: *Istnieje taki x, ze x jest Kubusiem i istnieje taki y, ze y jest Antykubusiem...*"

Teraz uwzględniamy stosunek panujący między pierwszą i drugą NAZWA, pamiętając, żeby zastosować ku temu symbol koniunkcji, gdyż ostatnim wpisanym przez nas kwantyfikatorem był mały kwantyfikator

$$\exists_x \{K(x) \wedge \exists_y [A(y) \wedge W(x, y)] \dots$$

co czytamy jako: *"Istnieje taki x, ze x jest Kubusiem i istnieje taki y, ze y jest Antykubusiem i x widział y..."* Kolejny krok to konieczność przedstawienia w schemacie kolejnego bohatera naszego zdania - czasu, który jest tu nierozłącznie związany z Antykubusiem - to on figluje z nim. Pamiętamy oczywiście o symbolu koniunkcji, łączącym istnienie tej NAZWY z tym, co dotąd napisaliśmy

$$\exists_x \{K(x) \wedge \exists_y [A(y) \wedge W(x, y) \wedge \exists_z (C(z) \dots$$

czytamy jako: *"Istnieje taki x, ze x jest Kubusiem i istnieje taki y, ze y jest Antykubusiem i x widział y i istnieje taki z, ze z jest czasem..."* No i nie pozostało nam nic innego, jak dopełnienie schematu relacją zachodzącą pomiędzy Antykubusiem i czasem - *"y gonił z"*, jak zwykle wpisując w odpowiednim miejscu symbol koniunkcji, bo determinuje to mały kwantyfikator:

$$\exists_x \{K(x) \wedge \exists_y [A(y) \wedge W(x, y) \wedge \exists_z (C(z) \wedge G(y, z))]\}$$

co czytamy jako: *Istnieje taki x, że x jest Kubusiem i istnieje taki y, że y jest Antykubusiem i x widział y i istnieje taki z, że z jest czasem i y gonił z."*

"Kubuś widział Antykubusia, goniącego czas."

[*"(Jeden) Kubuś widział (jednego) Antykubusia, goniącego (jeden) czas."*]

x - Kubuś

y - Antykubuś

z - czas

K(x) - x jest kubsiem

A(y) - y jest Antykubusiem

C(y) - z jest czasem

W(x, y) - x widział y

G(y, z) - y gonił z

$$\exists_x \{K(x) \wedge \exists_y [A(y) \wedge W(x, y) \wedge \exists_z (C(z) \wedge G(y, z))]\}$$

"Istnieje taki x, że x jest Kubusiem i istnieje taki y, że y jest Antykubusiem i x widział y i istnieje taki z, że z jest czasem i y gonił z."

❶ Wypisujemy zmienne nazwowe (NAZWY):

- x - mis
- y - miodek
- z - Człowiek

❷ Tworzymy zmienne predykatowe (PREDYKATY):

- $M(x)$ - x jest misiem
- $U(y)$ - y jest miodkiem
- $C(z)$ - z jest Człowiekiem
- $Z(x, y)$ - x zjada y
- $W(z, y)$ - z wyprodukował y

$$\forall x \{M(x) \rightarrow \sim \exists y [u(y) \wedge Z(x, y) \wedge \exists z (C(z) \wedge W(z, y))]\}$$

Negacja jest konieczna, ponieważ w naszym zdaniu jest jednoznaczne zaprzeczenie temu, że istnieje jakiś miodek ludzkiej produkcji, który odważyłby się zjeść wszystkie misie...

po kolei...

Rozpoczynamy od napisania faktu, że to, co tu dzieje się, dotyczy każdego misia:

$$\forall x \{M(x) \dots\}$$

UWAGA! Czyta się to tak: *"Dla każdego x , x jest misiem..."* Teraz kolejna NAZWA, która jest wobec misia podrzędną

$$\forall x \{M(x) \rightarrow \sim \exists y [u(y) \dots\}$$

"Dla każdego x , x jest misiem, to NIE istnieje taki y , że y jest miodkiem..."

Teraz relacja zachodząca między pierwszą i drugą NAZWA, pamiętamy, żeby zastosować symbol koniunkcji, gdyż ostatnim wpisanym przez nas kwantyfikatorem był mały kwantyfikator:

$$\forall x \{M(x) \rightarrow \sim \exists y [u(y) \wedge Z(x, y) \dots\}$$

Dla każdego x , x jest misiem, to NIE istnieje taki y , że y jest miodkiem i x zjada y ..."

Przedstawiamy w schemacie kolejnego bohatera naszego zdania - Człowieka, który jest tu nierozłącznie związany z miodkiem. Pamiętamy oczywiście o symbolu koniunkcji, łączącym istnienie tej NAZWY z tym, co dotychczas napisaliśmy (ostatnio wpisaliśmy mały kwantyfikator):

$$\forall x \{M(x) \rightarrow \sim \exists y [u(y) \wedge Z(x, y) \wedge \exists z (C(z) \dots\}$$

"Dla każdego x, x jest misiem, to NIE istnieje taki y, że y jest miodkiem i x zjada y, i istnieje taki z, że z jest Człowiekiem..." Dopełniamy schemat relacja zachodząca pomiędzy Człowiekiem i miodkiem - "z wyprodukował y", jak zwykle wpisując w odpowiednim miejscu symbol koniunkcji, bo determinuje to ostatni mały kwantyfikator:

$$\forall x \{M(x) \rightarrow \sim \exists y [u(y) \wedge Z(x, y) \wedge \exists z (C(z) \wedge W(z, y))]\}$$

"Dla każdego x, x jest misiem, to NIE istnieje taki y, że y jest miodkiem i x zjada y, i istnieje taki z, że z jest Człowiekiem, i z wyprodukował y."

Podsumowując, cała praca powinna wyglądać następująco: *"Wszystkie misie nie zjedzą miodku, wyprodukowanego przez Człowieka."* ["Dla każdego misia nie istnieje taki (jeden) miodek, który nadawałby się do zjedzenia i zostałby wyprodukowany przez (jednego) Człowieka ."]

x - mis

y - miodek

z - Człowiek

M(x) - x jest misiem

U(y) - y jest miodkiem

C(z) - z jest Człowiekiem

Z(x, y) - x zjada y

W(z, y) - z wyprodukował y

$$\forall x \{M(x) \rightarrow \sim \exists y [u(y) \wedge Z(x, y) \wedge \exists z (C(z) \wedge W(z, y))]\}$$

"Dla każdego x, jeżeli x jest misiem, to NIE istnieje taki y, że y jest miodkiem i x zjada y, i istnieje taki z, że z jest Człowiekiem i z wyprodukował y."

13.3.3 Zdanie „Istnieją ludzie, którzy są aniołami”

[mówiąc w uproszczeniu: "Istnieje taka (przynajmniej jedna) istota, która jest jednocześnie Człowiekiem i Aniołem."]

colorbluex - istota

C(x) - x jest Człowiekiem

A(x) - x jest Aniołem

$$\exists x (C(x) \wedge A(x))$$

- Mały kwantyfikator, bo zdanie nie mówi o wszystkich Ludziach, ale o niektórych z nich.

- Koniunkcja, bo to nieodłączna towarzyszka małego kwantyfikatora.

- W obu nawiasach "x", bo w tym przypadku chodzi o jedną i tą samą istotę, która jest jednocześnie Człowiekiem i Aniołem.

13.3.4 Zdanie „Istnieją ludzie, którzy nie są aniołami”

[mówiąc w uproszczeniu: "Istnieje taka (przynajmniej jedna) istota, która jest Człowiekiem i nie jest Aniołem."]

x - istota

$C(x)$ - x jest Człowiekiem

$A(x)$ - x jest Aniołem

$$\exists x(C(x) \wedge \sim A(x))$$

colorred" Istnieje taki x , że x jest Człowiekiem i x nie jest Aniołem."

13.3.5 Zdanie „Wszyscy ludzie są aniołami”

[mówiąc w uproszczeniu: "Każda istota, która jest Człowiekiem, jest jednocześnie Aniołem."]

x - istota

$C(x)$ - x jest Człowiekiem

$A(x)$ - x jest Aniołem

$$\forall x(C(x) \rightarrow A(x))$$

- Duży kwantyfikator, bo zdanie mówi o wszystkich Ludziach.
- Implikacja, bo to nieodłączna towarzyszka dużego kwantyfikatora.
- W obu nawiasach " x ", bo w tym przypadku chodzi o wszystkie i te same istoty, które są jednocześnie ludźmi i Aniołami.

colorred" Dla każdego x , jeżeli x jest Człowiekiem, to x jest Aniołem."

13.3.6 Zdanie „Żaden człowiek nie jest aniołem”

mówiąc w uproszczeniu:

- WARIANT I - "Każda istota, która jeżeli jest Człowiekiem, to nie jest Aniołem."
- lub też: WARIANT II - "Nie istnieje istota, która jest zarazem Człowiekiem i Aniołem."

x - istota

$C(x)$ - x jest Człowiekiem

$A(x)$ - x jest Aniołem

❶ *variant I*

$$\forall x(C(x) \rightarrow \sim A(x))$$

colorred" Dla każdego x , jeżeli x jest Człowiekiem, to x nie jest Aniołem."

❷ *variant II*

$$\sim \exists x(C(x) \wedge A(x))$$

colorred" Nie istnieje taki x , że x jest Człowiekiem i x jest Aniołem."

13.4 Znajdowanie schematów wnioskowania i dowody ich niezawodności

13.4.1 Przykład 1

Żadna ryba nie jest ssakiem

Żaden wieloryb nie jest rybą

Każdy wieloryb jest ssakiem

Jeśli założymy, że:

x - zwierzę

$R(x)$ - zwierzę jest rybą

$S(x)$ - zwierzę jest ssakiem

$W(x)$ - zwierzę jest wielorybem

to wówczas schemat ten z użyciem kwantyfikatorów mógłby wyglądać następująco:

$$\forall x (R(x) \rightarrow \neg S(x))$$

$$\forall x (W(x) \rightarrow \neg R(x))$$

$$\forall x (W(x) \rightarrow S(x))$$

Teraz następuje proces ujednoczenia kwantyfikatorów, w celu ich usunięcia z zapisu (w naszym konkretnym przypadku wszystkie kwantyfikatory są identyczne, więc można je usunąć swobodnie):

$$R(x) \rightarrow \neg S(x)$$

$$W(x) \rightarrow \neg R(x)$$

$$W(x) \rightarrow S(x)$$

Stosując metodę skróconą zerojedynekową łatwo przekonujemy się, iż schemat ten jest **niezawodny**.

Przystępujemy więc do dowodu jego niezawodności. Uproszczony zapis schematu to:

$$r \rightarrow \neg s$$

$$w \rightarrow \neg r$$

$$w \rightarrow s$$

• metodą założeniową <i>wprost</i>	• metodą założeniową <i>niewprost</i>
1 $r \rightarrow \neg s$ (zał.)	1 $r \rightarrow \neg s$ (zał.)
2 $w \rightarrow \neg r$ (zał.)	2 $w \rightarrow \neg r$ (zał.)
3 w (zał.)	3 $\neg(w \rightarrow s)$ (DN)
4 $\neg r$ RO(2,3)	4 $\neg(\neg w \vee s)$ ZI(3)
5 $r \wedge \neg\neg s$ ZI(1)	5 $\neg\neg w \wedge \neg s$ NA(4)
6 $r \wedge s$ PN(5)	6 $w \wedge \neg s$ PN(5)
7 s OK(6)	7 $\neg s$ OK(6)
	8 w OK(6)
	9 $\neg r$ RO(2,8)
	10 $r \wedge \neg\neg s$ ZI(1)
	11 $r \wedge s$ PN(10)
	12 s OK(11)

SPRZECZNE 7 i 12.

13.4.1 Przykład 2

Nie każdy człowiek jest pijakiem

Każdy pijak jest człowiekiem

nie każdy człowiek jest człowiekiem

Jeśli założymy, że:

x - istota

$C(x)$ - istota jest człowiekiem

$P(x)$ - istota jest pijakiem

to wówczas schemat ten z użyciem kwantyfikatorów mógłby wyglądać następująco:

$\neg \forall x (C(x) \rightarrow P(x))$

$\forall x (P(x) \rightarrow C(x))$

$\neg \forall x (C(x) \rightarrow C(x))$

Teraz następuje proces ujednoczenia kwantyfikatorów, w celu ich usunięcia z zapisu. W tym celu należy zamienić kwantyfikatory ogólne na kwantyfikatory szczegółowe .

Teraz schemat wygląda następująco:

$\exists x \neg (C(x) \rightarrow P(x))$

$\exists x (P(x) \rightarrow C(x))$

$\exists x \neg (C(x) \rightarrow C(x))$

Teraz wszystkie kwantyfikatory są takie same, więc można je usunąć:

$\neg (C(x) \rightarrow P(x))$

$P(x) \rightarrow C(x)$

$\neg (C(x) \rightarrow C(x))$

Stosując metodę skróconą zerojedynekową łatwo przekonujemy się, iż schemat ten jest **niezawodny**.

Przystępujemy więc do dowodu jego niezawodności.

Uproszczony zapis schematu to:

$\neg (c \rightarrow p)$

$p \rightarrow c$

$\neg (c \rightarrow c)$

Zgodnie z metodą założeniową *nie wprost* dowód wygląda następująco:

1 $\neg (c \rightarrow p)$ (zał.)

2 $p \rightarrow c$ (zał.)

3 $(c \rightarrow c)$ (DN)

4 $c \wedge \neg c$ ZI(3)

5 c OK(4)

6 $\neg c$ OK(4)

SPRZECZNE 5 i 6, zatem schemat jest niezawodny.

14. Wiedza niepewna:

Wiedzą niepewną będziemy określać taką wiedzę, której ekspert tę wiedzę przekazujący ufa w większej części, i zakłada, że w większości przypadków ta wiedza się sprawdza w rzeczywistości. Jednak nie ma on 100% przekonania o tym, że będzie ona prawdziwa w każdej sytuacji.

Pojęcia nieostre

Pojęcia nieostre występują zawsze wtedy gdy wiedza zapisana jest przy użyciu pojęć typu: "stan pacjenta stabilny" czy "odpowiednia dawka leku". Bez odpowiedniego aparatu matematycznego wspomagającego tak zapisaną wiedzę np. w postaci współczynników pewności czy np. probabilistyki, wnioskowanie w takim systemie jest niemożliwe

Pojęcia niespójne

Niepewność objawia się w ten sposób np., że przy takich samych warunkach w danej bazie wiedzy mamy reguły o innych decyzjach, które uniemożliwiają podjęcie jednoznacznej decyzji. Ten rodzaj niepewności wiedzy rozwiązują doskonale *zbiory przybliżone*

Niepewność może występować zarówno w faktach jak i w regułach. Do rozwiązania problemu niepewności w bazach wiedzy wykorzystuje się:

- **prawdopodobieństwo zajścia jakiegoś zdarzenia (faktu),**
- **zbiory rozmyte,**
- **współczynnik CF,**
- **teoria Dempstera-Sheffera,**
- **zbiory przybliżone,** gdzie wiedza pewna jest określona przez dolne lub górne przybliżenie zbioru, a to, co znajduje się na brzegu reprezentuje wiedzę niepewną (brzeg to różnica między górnym a dolnym przybliżeniem zbioru).

14.1. Zbiory rozmyte:

Metoda reprezentacji wiedzy wyrażonej w języku naturalnym:

Temperatura wynosi 29 °C

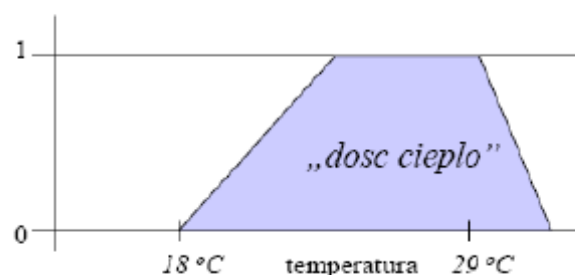


Jest dość ciepło

informacja liczbowo - naturalna dla systemów komputerowych

informacja opisowa - naturalna dla człowieka

Zamiast dwóch wartości logicznych (prawda i fałsz), dopuszcza się istnienie nieskończenie wielu wartości (odpowiadających liczbom rzeczywistym od 0 do 1).

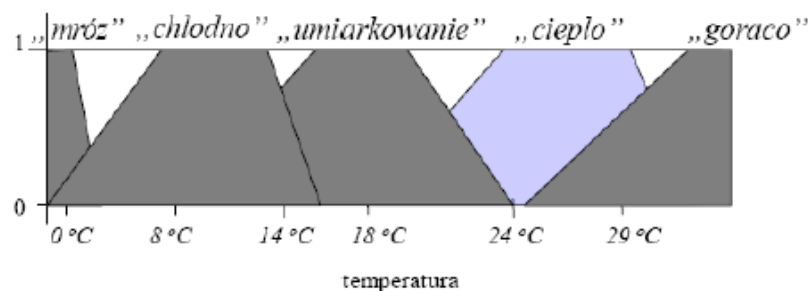


W klasycznej teorii zbiorów obowiązują m.in. dwa prawa:

- prawo niesprzeczności
- prawo wyłączonego środka.

Inaczej mówiąc, każdy element należy albo do zbioru, albo do jego dopełnienia. Nie może należeć do obu naraz. Jeśli mamy np. pojęcia: dzień i noc, to one się wzajemnie wykluczają. Temperatura otoczenia może być tylko albo ujemna, albo nieujemna. W teorii zbiorów rozmytych przyjmuje, że element może należeć częściowo do zbioru jak i do jego dopełnienia. Stopień przynależności elementu x do zbioru A określa **funkcja przynależności**, oznaczana zwykle $\mu_A(X)$, o wartościach w przedziale $[0, 1]$. Zbiory rozmyte opisują najczęściej pojęcia lingwistyczne używane często w życiu codziennym jak np. chłodno, gorąco.

Zastosowanie



Pojęcia „ciepło” czy „gorąco” są określone w sposób nieostry: trudno jednoznacznie określić ich granice, ich zakresy mogą się częściowo pokrywać.

Zbiory rozmyte pozwalają konstruować reguły typu:

jeśli temperatura jest wysoka i wilgotność jest niska, to sasiad biega

Reguły rozmyte

Reguły, których przesłanki lub wnioski wyrażone są w języku zbiorów rozmytych.

*Jeżeli x jest **male** i y jest **srednie**, to uruchom alarm.*

*Jeżeli x jest **male** i y jest **male**, to ustaw z na **duze**.*

*Jeżeli x jest **duze**, to ustaw z na **male**.*

Reguły pochodzące od ekspertów zwykle wyrażone są w języku nieprecyzyjnym. Zbiory rozmyte pozwalają nam przelożyć ten język na konkretne wartości liczbowe.

14.2 Sieci Bayesa:

G to graf określony zbiorem wierzchołków N i krawędzi E . CP to zbiór prawdopodobieństw warunkowych opisujących prawdopodobieństwo przejścia od jednego wierzchołka grafu do drugiego.

Pod pojęciem **sieci Bayesowskiej** rozumiemy trójkę: $B = \{N, E, CP\}$, gdzie dwójka $\{N, E\}$ jest zorientowanym grafem acyklicznym zbudowanym na podstawie zadanych **prawdopodobieństw warunkowych** zawartych w zbiorze CP .

Sieć bayesowska służy do przedstawiania zależności pomiędzy zdarzeniami bazując na rachunku prawdopodobieństwa. Klasycznym przykładem jest reprezentowanie zależności pomiędzy **symptomami** a chorobą. Formalnie taka sieć jest modelowana za pomocą **skierowanego grafu acyklicznego**, w którym wierzchołki reprezentują zdarzenia, a łuki związki przyczynowe pomiędzy tymi zdarzeniami. Jeśli od wierzchołka A prowadzi ścieżka do wierzchołka B to B jest potomkiem A . Podstawowym założeniem sieci bayesowskiej jest niezależność danego zdarzenia od wszystkich innych, które nie są jego potomkami.

Prawdopodobieństwo warunkowe to prawdopodobieństwo zajścia zdarzenia A pod warunkiem zdarzenia B - co odpowiada prostej regule "Jeżeli B to A ", którego ogólna postać wygląda następująco:

$$P(A/B) = \frac{P(B/A) * P(A)}{P(B)}$$

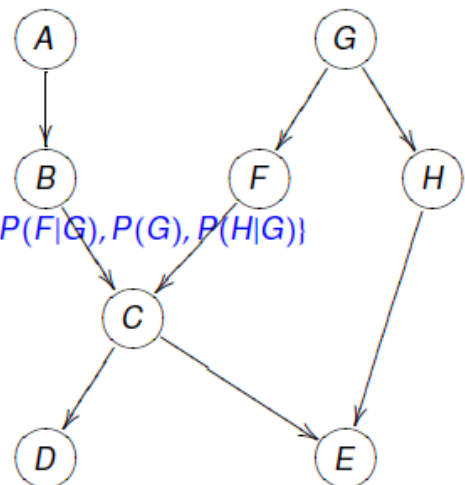
i oznacza, że stwierdzenie A może być uznane jako prawdziwe wtedy, kiedy stwierdzenie B jest uznane jako prawdziwe.

Niech zbiór pewnych zmiennych identyfikujących obserwacje i hipotezy ma następującą postać:

$Z = \{A, B, C, D, E, F, G, H\}$,

$CP = \{P(A), P(B|A), P(C|B), P(C|F), P(D|C), P(E|CH), P(F|G), P(G), P(H|G)\}$

To pozwala zbudować graf skierowany, który opisuje sieć Bayesa: $B = \{N, E, CP\}$, co można przedstawić graficznie:



Sieć Bayesa stanowi numeryczny model związków przyczynowo-skutkowych zachodzących pomiędzy elementami zbioru obserwacji i hipotez. Możliwe jest wówczas wnioskowanie progresywne (w przód), jak i wnioskowanie regresywne (wstecz).

Dostosowując się do wzoru Bayes'a, w przypadku, gdy mamy dwa fakty:

A - jeżdżę na rowerze,

oraz B - jest ładna pogoda,

gdzie $P(A) = 0,2$ i $P(B) = 0,4$

oraz równocześnie w bazie wiedzy istnieją reguły :

$R1$: Jeżeli jest ładna pogoda to jeżdżę na rowerze- co po prostu oznacza $P(A/B)$

$R2$: Jeżeli jeżdżę na rowerze to jest ładna pogoda- co odpowiednio oznacza $P(B/A)$,

to znając prawdopodobieństwo zajścia zdarzenia B pod warunkiem A , tzn., gdy wiemy, że $P(B/A) = 0,8$, możemy także określić prawdopodobieństwo zajścia zdarzenia A pod warunkiem B . Korzystając z wzoru Bayes'a otrzymujemy wartość $P(A/B) = [(0,8 * 0,4) / 0,2] = 0,4$. Wzór ten pozwala nam ustalić pewną hipotezę pod warunkiem, że znamy hipotezę przeciwną.

14.3 Współczynniki pewności:

Metoda współczynników pewności zakłada rozszerzenie modelu regułowego o pewne numeryczne oszacowanie stopnia pewności eksperta o prawdziwości danej reguły czy też faktu.

$\langle \textit{student}, \textit{srednia_ocen}, \textit{wysoka}, CF = 0.5 \rangle$

oznaczać ma po prostu fakt, że dany student ma przyznane stypendium ze stopniem pewności $CF = 0.8$. Wielkość ta ma określać stopień naszego przekonania o prawdziwości konkluzji danej reguły w przypadku prawdziwości jej przesłanki.

rozszerzona reguła może mieć następującą postać:

Jeżeli $e_1 \& e_2 \& \dots \& e_n$ To h ze stopniem pewności CF

gdzie e_1, e_2, \dots, e_n to przesłanki reguły a h to konkluzja, $\&$ to operator logiczny And.

Zatem zapis stwierżeń niepewnych (hipotez, przypuszczeń), uzupełniający każdą trójkę $\langle O, A, V \rangle$ o stopień pewności CF (ang.: *Certainty Factor*), powoduje, że ostatecznie ta metoda reprezentacji wiedzy ma postać czwórki: $\langle O, A, V, CF \rangle$.

W systemach zbliżonych do modelu MYCIN wnioskowanie odbywa się w sposób klasyczny, z wykorzystaniem interpretera reguł produkcji, który np. w systemie MYCIN pracuje w trybie wnioskowania wstecz. W trakcie tego procesu niepewność jest uwzględniana w kolejnych krokach wnioskowania poprzez obliczenie współczynnika pewności poszczególnych konkluzji.

Proces ten ma jednak charakter pomocniczy i to nie on steruje procesem wnioskowania, główną rolę odgrywa tutaj interpreter reguł. Innymi słowy, przetwarzanie niepewności jest tutaj procesem równoległym, mającym na celu określenie stopnia pewności konkluzji generowanych przez interpreter reguł.

Przypomnijmy, że w systemach Bayes'owskich (i podobnych) to mechanizm przetwarzania wiedzy niepewnej decydował o konkluzji i określał pewne numeryczne oszacowanie jej pewności (w postaci prawdopodobieństw czy np. Dempster'owko-Shafer'owskich mas).

Do odwzorowania wiedzy służy współczynnik MB zwany *miarą wiarygodności* (ang. *measure of belief*), do opisanie niewiedzy służy zaś współczynnik MD zwany *miarą niewiarygodności* (ang. *measure of disbelief*). Ponieważ współczynnik CF wiązany jest z regułą, również współczynniki MB i MD są związane z regułą. Załóżmy, że dana jest reguła:

Jeżeli e to h .

Współczynniki dla takiej reguły będą określone odpowiednio $MB(h, e)$, $MD(h, e)$, $CF(h, e)$. Współczynnik $CF(h, e)$ jest zdefiniowany jako różnica pomiędzy miarą wiarygodności a miarą niepewności:

$$CF(h, e) = MB(h, e) - MD(h, e)$$

Interpretacja miar wiarygodności i niewiarygodności (w powiązaniu z prawdopodobieństwem warunkowym) może być następująca:

- jeżeli $P(h | e) = 1$ to h jest prawdziwe na pewno, wtedy $MB(h, e) = 1$, $MD(h, e) = 0$, oraz $CF(h, e) = 1$,
- jeżeli $P(\sim h | e) = 1$ to h jest fałszywe na pewno, wtedy $MB(h, e) = 0$, $MD(h, e) = 1$, oraz $CF(h, e) = -1$,
- jeżeli $P(h | e) = P(h)$ to h co znaczy, że h i e są niezależne, wtedy $MB(h, e) = 0$, oraz $MD(h, e) = 0$, $CF(h, e) = 0$.

1) szeregowo:

$$\textcircled{A} \rightarrow \textcircled{B} \rightarrow \textcircled{C} \quad ; \quad \textcircled{AB} \rightarrow \textcircled{C}$$

$$CF(C|AB) = CF(BA) \cdot CF(C|B)$$

b) równoległe:

The diagram shows two nodes, A and B, with arrows pointing to a central node C. To the right of this, an arrow points from a node labeled AB to node C. An equivalence symbol (⇒) is placed between the two diagrams.

$$\begin{array}{l} (+) \quad CF(C|A) + CF(C|B) - CF(C|A)CF(C|B) \\ (-) \quad CF(C|A) + CF(C|B) + CF(C|A)CF(C|B) \\ (+, -) \quad \frac{CF(C|A) + CF(C|B)}{1 - \min\{|CF(C|A)|, |CF(C|B)|\}} \end{array}$$

ZALETY

- Prostota i łatwość w interpretacji,
- Powiązanie z najbardziej popularną reprezentacją wiedzy w postaci reguł produkcji,
- Stosunkowo łatwe obliczenia nie obciążające czasowo ani pamięciowo.

WADY

- Mało stabilna podbudowa teoretyczna,
- Bardzo luźny związek z teorią prawdopodobieństwa,
- Udowodniono wyraźne rozbieżności pomiędzy wynikami wnioskowania czysto probabilistycznego a w oparciu o model CF,
- Pojedynczy współczynnik CF jest zbyt słabym narzędziem do odwzorowania wiedzy i niewiedzy.