ZESZYTY NAUKOWE POLITECHNIKI SLASKIEJ 1992
Seria: AUTOMATYKA z.109 Nr kol. 1175

Zbigniew Bartoń, Bożena Paluchiewicz
Politechnika Śląska

O PEWNYM MODELU OBLICZENIOWO - LOGICZNYM REPREZENTACJI WIEDZY

ABOUT SOME CALCULATED - LOGICAL MODEL OF REPRESENTATIVE KNOWLEDGE

О НЕКОТОРОЙ ЛОГИЧЕСКО - ИСЧИСЛЕННОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Streszczenie: W artykule przedstawiono podstawy matematyczne tworzenia modeli bazy wiedzy. Opis wiedzy oparto na modelu obliczeniowo - logicznym. Wykorzystanie zaproponowanego podejścia może mieć zastosowanie m.in. w robotyce oraz w automatyzacji systemów przemysłowych.

Summary: In this paper basic construction of model base knowledge are presented. Application of this method may be used in the robotic and automatic control system industry.

Резюме: В статье представлено основы определения математических модели знании. Эти модели численно - логические возможно использовать в роботике и автоматизации промышленных систем.

1. Wprowadzenie

Podstawowym zagadnieniem przy tworzeniu bazy wiedzy jest znalezienie sposobu jej przedstawienia tj. opisania wiedzy i zapewnienia pełnego kontaktu z użytkownikiem. Wybór sposobu zapisywania wiedzy uzależniony jest od wielu czynników, z których do najważniejszych należy zaliczyć rodzaj wiedzy wymaganej do poprawnego działania systemu oraz niezbędną wielkość bazy wiedzy. Podział form prezentacji wiedzy na deklaratywne i proceduralne ma charakter względny, gdyż konkretne modele w różnym stopniu wykorzystują obie formy prezentacji. Najbardziej rozpowszechnione stały się logiczne oraz sieciowe modele reprezentacji wiedzy [1, 2, 3, 4, 5, 6].

Istotą modeli logicznych jest pojęcie formalnego systemu zadeklarowanego przez cztery symbole M = (T, P, A, F), gdzie:

T - zbiór podstawowych (bazowych) elementów,

P - zbiór reguł syntaktycznych pozwalających zbudować z T syntaktycznie poprawne wyrażenie,

A - zbiór, a priori rzeczywistych aksjomatów,

F - semantyczne zasady wyjść, pozwalające rozszerzyć zbiór aksjomatów, dzięki dodatkowym wyrazeniom.

Wykorzystanie róznego typu logik przy tworzeniu syntaktycznych i semantycznych zasad pozwala na budowę róznego typu modeli logicznych. Najszerzej wykorzystuje się modele oparte na obliczaniu predykatów.

2. Zmienne zdaniowe

W logice matematycznej zmienną zdaniową nazywamy każdą zmienną, której wartosciami mogą być dowolne zdania, prawdziwe lub fałszywe.

Załóżmy, że zmiennym zdaniowym będziemy przyporządkowywać łacińskie litery bez indeksów A, B, C,..., które nazwiemy formułami rachunku zdań. W związku z tym, że zmienne zdaniowe mogą być prawdziwe lub fałszywe formuły rachunku zdań mogą przyjmować wartości Y (prawda) lub N (fałsz). Jednakże, w szczególności w teorii automatów skończonych i logice matematycznej prawdzie przyporządkowuje się 1 a fałszowi 0.

Bardziej złożone formuły rachunku zdań można uzyskać wykorzystując rzeczywiste - funkcjonalne zależności F(A,B,...), które są określane przez rzeczywiste wartości A, B, ...Funkcję F(A,B,...) zazwyczaj deklaruje się w postaci tablicy zwanej tablicą rzeczywistą funkcji rzeczywistej.

Funkcją rzeczywistą lub funkcją algebry logiki nazywa się każdą funkcję zawierającą n rzeczywistych argumentów, która przyjmuje wartości Y (1) lub N (0).

3. Operacje logiczne

Zmienne zdaniowe można za pomocą operacji logicznych łączyć w złożone formuły rachunku zdań. Do podstawowych operacji logicznych zaliczamy:

- negację, inaczej dopełnienie, oznaczane symbolami $\bar{A}$, $\neg A$, $\sim A$, $-A$,
- koniunkcję, inaczej iloczyn logiczny, oznaczane symbolami $A \wedge B$, AB, $A\&B$
- alternatywę, inaczej sumę logiczną, oznaczane symbolami $A \vee B$, $A+B$,
- implikację, oznaczoną symbolami $A \Rightarrow B$, $A \supset B$,
- równoważność, inaczej ekwiwalentność, oznaczane symbolami $A \Leftrightarrow B$, $A \equiv B$.

4. Formuły rachunku zdań

Formuła rachunku zdań jest to dowolny napis utworzony ze zmiennych zdaniowych i operacji logicznych (spójników logicznych), który po każdym podstawieniu zdań na miejsce wszystkich zmiennych zdaniowych zmienia się w zdanie (prawdziwe lub fałszywe). Zakładamy przy tym, ze w każdym podstawieniu jednakowe zmienne zastępowane są tym samym zdaniem. Formuły rachunku zdań noszą także nazwy formuł algebry logiki lub formy propozycyjnej. Zauważmy, że:

- każda zmienna zdaniowa jest formułą rachunku zdań,
- jeżeli F i G są formułami rachunku zdań, to F, $(F \vee G)$, $(F \wedge G)$ $(F \Rightarrow G)$ oraz $(F \equiv G)$ są także formułami rachunku zdań,
- żadne inne wyrazenie nie jest formułą rachunku zdań.

Dla prostoty formuły rachunku zdań będą dalej nazywane formułami. Każda formuła określa funkcję rzeczywista.

Dwie różne formuły zadające jedna i tę samą funkcję rzeczywistą nazywaja się równoważnymi. Równoważność formuł F i G zapisujemy F = G.

Formuła rachunku zdań, która jest prawdą przy dowolnych rzeczywistych wartościach jej argumentów (przy każdym podstawieniu zdań w miejsce wszystkich zmiennych zdaniowych zmienia sie w zdanie prawdziwe) nazywa się tautologia.

Dwie formuły rachunku zdań F i G nazywamy równoważnymi wtedy i tylko wtedy, gdy formuła F = G jest tautologią. Przykładem tautologii są: $(A \vee \bar{A})$ "prawo wykluczania trzeciego", $(A \equiv A)$, $\overline{(A \wedge \bar{A})}$.

Formuła, która jest fałszywa dla każdej rzeczywistej wartości argumentow nazywa sie sprzecznością. Przykładami sprzeczności są: $(A \equiv \bar{A})$, $(A \wedge \bar{A})$.

Rozpatrywane operacje logiczne nie są niezależne, tzn. jedne z nich można wyrazić przez drugie. W szczególności operacje logiczne implikację i równoważność można wyrazić przez pozostałe. Łatwo sprawdzić wykorzystując rzeczywiste tablice, że formuła:

$$(A \Rightarrow B) = (\bar{A} \vee B),$$

a $$(A \equiv B) = ((\bar{A} \vee B) \wedge (\bar{B} \vee A)).$$

Dlatego też, w dalszej części zwrócimy uwagę na właściwości trzech operacji logicznych: negacji, koniunkcji i alternatywy.

4.1. Podstawowe właściwości wybranych operacji logicznych

Metody badania formuł, oparte na bezpośrednim sprawdzaniu fałszywości lub prawdy, są nazywane w logice semantycznymi, w odróżnieniu od metod syntaktycznych, polegających na przekształcaniu formuł zgodnie z pewnymi regułami, tak aby doprowadzić do pewnej pożądanej postaci. Badając metodami semantycznymi formuły, nie musimy podstawiać zamiast zmiennych zdaniowych konkretnych zdań, bo i tak jedyną interesującą cechą tych zdań byłaby prawdziwość. Wystarczy więc podstawiać zamiast zmiennych zdaniowych wartości 1 (prawda) lub 0 (fałsz).

Łatwo udowodnić następujące właściwości operacji logicznych:

a) łączność operacji koniunkcji i alternatywy,

b) przemienność operacji koniunkcji i alternatywy,

c) idempotentność koniunkcji i alternatywy

$$A \wedge A = A$$
$$A \vee A = A,$$

d) rozdzielność operacji koniunkcji i alternatywy,

e) także

$$A \wedge \bar{A} = 0 \quad A \vee \bar{A} = 1 \quad A \wedge 0 = 0 \quad A \wedge 1 = A$$
$$A \vee 0 = 0 \quad A \vee 1 = 1 \quad \bar{\bar{A}} = A,$$

f) jeżeli przyjąć, że

$$A^0 = \bar{A}.$$
$$A^1 = A$$

i σ_j przy dowolnym $j = 1, 2, \ldots, n$ są równe 0 lub 1, to dowolna koniunkcja

$$A_1^{\sigma_1} \wedge \ldots \wedge A_n^{\sigma_n} = 1$$

zachodzi wtedy i tylko wtedy, gdy $A_i = \sigma_i$, $i = 1, \ldots, n$.

Przytoczone wyzej właściwości a) - f) pozostają prawdziwe przy zamianie zmiennych zdaniowych dowolną formułą rachunku zdań,

g) jeżeli $F(A, B, \ldots)$ - jest formułą zawierającą tylko operacje logiczne negacji, koniunkcji i alternatywy, to dla otrzymania negacji $\bar{F}(A, B, \ldots)$ w formie wyjściowej $F(A, B, \ldots)$ konieczne jest wszystkie argumenty zamienić ich negacjami, a operacje koniunkcji i alternatywy zamienić miejscami, tj. symbol $\wedge$ zamienić na symbol $\vee$ i odwrotnie $\vee$ na $\wedge$. Cechę tę nazywa się prawem de Morgana.

5. Formuły normalne

Wykorzystując przytoczone właściwości operacji logicznych, można stworzyć szereg twierdzeń, które będą stosowane przy przekształcaniu funkcji rzeczywistych.

Twierdzenie 5.1

Dowolną formułę $F(A_1, A_2, \ldots A_{k+1}, \ldots, A_m)$, z wyjątkiem formuły będącej negacją (tj. równej tożsamościowo zero), można przedstawić w postaci:

$$F(A_1,\ldots,A_k,A_{k+1},\ldots,A_m) = \bigvee_{\sigma_1\ldots\sigma_k} A_1^{\sigma_1},\ldots,A_k^{\sigma_k}\ F_k(\sigma_1,\ldots,\sigma_k,A_{k+1},\ldots,A_m), \tag{5.1}$$

gdzie: - symbol $\bigvee_{\sigma_1\ldots\sigma_k}$ oznacza sumę logiczną po całym zbiorze $(\sigma_1,\ldots,\sigma_k)$;

- σ_i $(i = 1,\ldots, m)$ w każdym zbiorze przyjmuje wartości 0 lub 1.

Przedstawienie rzeczywistej funkcji w postaci (5.1) nazywa się rozkładem funkcji względem k zmiennych.

Twierdzenie 5.2

Każdą formułę $F(A_1,\ldots, A_m)$, z wyjątkiem formuły będącej negacją, można przedstawić w postaci (5.2):

$$F(A_1,\ldots, A_m) = \bigvee_{F(\sigma_1,\ldots,\sigma_m)} A_1^{\sigma_1},\ldots, A_m^{\sigma_m}. \tag{5.2}$$

Po prawej stronie (5.2) sumowanie logiczne przeprowadza się po całym zbiorze $\sigma_1,\ldots, \sigma_m$, dla którego $F(\sigma_1,\ldots, \sigma_m) = 1$.

Prawa strona równania (5.2) nazywa się idealną alternatywną formułą normalną (IAFN) funkcji $F(A_1,\ldots,A_m)$.

Człony typu $(A_1^{\sigma_1},\ \ ,A_m^{\sigma_m})$, wchodzące w skład (IAFN), nazywają się alternatywnymi członami danej IAFN. W ten sposób funkcja IAFN przedstawia logiczną sumę pewnej liczby alternatywnych członów.

Z relacji (5.2) wynika sposób tworzenia IAFN zadanej w postaci tabelarycznej. Sposób ten można opisać następująco:

- z tablicy należy wybrać zbiory $(\sigma_1,\ldots,\sigma_m)$, tj. wiersze z 0 i 1, dla których $F(A_1,\ldots,A_m) = 1$,
- następnie dla każdego zbioru kompletuje się alternatywny człon $(A_1^{\sigma_1},\ldots,A_m^{\sigma_m})$ i wszystkie takie człony łączy się znakiem alternatywy.

Niech np. rzeczywista funkcja $F(A_1, A_2, A_3)$ ma postać jak w tablicy 1. Należy utworzyć IAFN tej funkcji.

Tablica 1

Nr wiersza	A_1	A_2	A_3	F
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	1
8	1	1	1	0

Zgodnie z wyżej przedstawionym sposobem tworzenia IAFN wybieramy wiersze 2 4 i 7: (0, 0, 1), (0, 1, 1), (1, 1, 0) i wg nich tworzymy IAFN zadanej funkcji.

$$F_1(A_1, A_2, A_3) = (A_1^0\ A_2^0\ A_3^1) \vee (A_1^0\ A_2^1\ A_3^1) \vee (A_1^1\ A_2^1\ A_3^0) =$$
$$= (\bar{A}_1\ \bar{A}_2\ A_3) \vee (\bar{A}_1\ A_2\ A_3) \vee (A_1\ A_2\ \bar{A}_3).$$

Czasami wygodniej jest, aby rzeczywista funkcja była przedstawiona w formie logicznego wyrażenia pewnej liczby członów typu: $(A_1^{\sigma_1} \vee \ldots \vee A_k^{\sigma_k} \vee \ldots \vee A_k^{\sigma_m})$. Przedstawienie funkcji w taki sposób nazywa sie idealną koniunkcyjną formułą normalną (IKFN).

Człony postaci $(A_1^{\sigma_1} \vee A_2^{\sigma_2} \vee \ldots \vee A_m^{\sigma_m})$ wchodzące do IKFN nazywamy członami koniunkcyjnymi.

Twierdzenie 5.3

Dowolną rzeczywistą formułę, z wyjątkiem formuły będącej tautologią, można przedstawić w postaci idealnej koniunkcyjnej formuły normalnej.

6. Zasady przedstawiania tautologii i przeczeń

Rozważmy niektóre z zasad tworzenia tautologii i przeczeń z alternatywnych i koniunkcyjnych członów.

1. Iloczyn logiczny dwóch różnych alternatywnych członów, zawierających m elementów, jest przeczeniem (tożsamościowo równy 0) wtedy i tylko wtedy, jeśli jeden z członów zawiera negację co najmniej jednego z argumentów wchodzących w skład drugiego alternatywnego członu:
$$(A_1\ A_2\ A_3)\ (\bar{A}_1\ A_4\ A_5) = 0.$$
2. Suma logiczna dwóch różnych koniunkcyjnych członów zawierających m argumentów jest tautologią wtedy i tylko wtedy, jeśli jeden z jej członów zawiera negację co najmniej jednego z argumentów wchodzących w skład drugiego koniunkcyjnego członu:
$$(A_1 \vee A_2 \vee \bar{A}_3) \vee (A_4 \vee A_5 \vee A_3) = 1.$$
3. Suma logiczna wszystkich 2^m parami różnych alternatywnych członów m argumentów jest tautologią.
$$\bigvee_{\sigma_1,\ldots,\sigma_m} A_1^{\sigma_1} \ldots A_m^{\sigma_m} = 1$$
4. Iloczyn logiczny 2^m parami różnych członów koniunkcyjnych m argumentów jest przeczeniem.
$$\bigwedge_{\sigma_1,\ldots,\sigma_m} (A_1^{\sigma_1} \vee \ldots \vee A_m^{\sigma_m}) = 0$$

Symbol $\bigwedge_{\sigma_1,\ldots,\sigma_m}$ oznacza iloczyn logiczny po całym zbiorze $(\sigma_1,\ldots,\sigma_m)$.

7. Synteza automatów skończonych

Aparat obliczeniowy algebry logiki jest szeroko stosowany do syntezy i analizy automatów skończonych. W tych przypadkach wartości zmiennych zdaniowych oznaczają stany elementów przełączających. Zazwyczaj przyjmuje się, że wartość zmiennej zdaniowej równa jest 1, jeżeli odpowiada jej styk zamknięty oraz 0, gdy styk jest otwarty. Przykłady wykorzystania przedstawionej teorii do syntezy systemów przełączających można znaleźć w literaturze dotyczącej teorii automatów skończonych.

8. Rachunek predykatów

W rozważaniach, w których mamy do czynienia z pewnymi zbiorami obiektów, możemy używać nie tylko zdań, które mogą być prawdziwe lub fałszywe, ale także funkcji zdaniowych, czyli predykatów.
Predykatem lub inaczej funkcją zdaniową (logiczną) nazywamy funkcję o dowolnej liczbie argumentów przyjmującą rzeczywiste wartości: Y lub N (1 lub 0).
Predykaty służą do oznaczania własności obiektów lub związków pomiędzy obiektami. Argumenty przyjmują wartości z dowolnego skończonego lub nieskończonego zbioru, który nazywa się obszarem nazwowym. Predykat o n argumentach nazywany jest n warunkowym predykatem.
Jeżeli $F(x)$, $G(x,y)$, $P(x_1, \ldots, x_n)$ są predykatami, to formuły F, G i P nazywają się formułami rachunku predykatów, natomiast argumenty x, y, x_1, ..., x_n zmiennymi nazwowymi. Często mówi się, że predykat $F(x)$ otrzymano w efekcie przyporządkowania formuły rachunku predykatów do zmiennej nazwowej.
Zamiast zmiennych nazwowych do predykatów mogą być podstawiane określone wielkości z obszaru nazwowego M - stałe nazwowe, a także n-wymiarowe funkcje $f(x_1,\ldots,x_n)$ odwzorowujące M^n w M, tj. funkcje przyjmujące wartości z M i określone w M^n. O funkcji $f(x_1,\ldots,x_n)$ mówi się zazwyczaj, że została w efekcie przypisana n-wymiarowej formule rachunku predykatów zmiennych nazwowych $x_1,\ldots,x_n$.
Predykat $F(x)$ określony w obszarze nazwowym M deklaruje określone właściwości elementów zbioru M, a interpretowany jest jako informacja "x posiada własność F", przy czym $F(x)$ przyjmuje wartość Y, jeżeli informacja jest prawdą oraz N, jeśli jest fałszem.
Predykat $F(x_1,\ldots,x_n)$ określa związki między elementami $x_1,\ldots,x_n$ i interpretuje się go jako określenie informacji "$x_1,\ldots,x_n$ znajdują się między sobą w zalezności F".
Niech np. M jest zbiorem liczb naturalnych. Wtedy predykat $F(x)$ może oznaczać, że "x jest liczbą parzystą" lub "x jest liczbą pierwszą". Predykat $G(x,y)$ może oznaczać: "x jest większe od y" lub "x nie jest większe od y" itd.

9. Kwantyfikatory

Rzeczywiste funkcje są szczególnym przypadkiem predykatów wtedy, gdy ich argumenty przyjmują tylko dwie wartości 0 lub 1. Wszystkie operacje na zmiennych zdaniowych można przenieść do rachunku predykatów i wykorzystać w tym celu związki między predykatami i formułami.

W logice predykatów dla spójnego zapisu wiadomości typu "dla dowolnego x istnieje F(x)" i "istnieje takie x, dla którego F(x)" wprowadza się dwie nowe operacje - kwantyfikator ogólny $\forall$ oraz kwantyfikator istnienia $\exists$. Przy wykorzystaniu tych operacji przedstawione powyżej informacje można zapisać w postaci:

$\forall x\ F(x)$ i odpowiednio $\exists x\ F(x)$.

Zmienną stojącą w formule rachunku predykatów bezpośrednio za znakiem kwantyfikatora nazywamy zmienną objętą tym kwantyfikatorem. Mówimy, że kwantyfikator wiąże zmienną nazwową x, jeżeli ona występuje pod tym kwantyfikatorem i w jego zasięgu. Zmienne nazwowe, występujące w formule rachunku predykatów, które nie są związane żadnym kwantyfikatorem, nazywamy zmiennymi wolnymi.

10. Interpretacja

Formuła ma określony sens, tj. oznacza określoną informację, jeżeli istnieje jakakolwiek interpretacja. Interpretować formułę znaczy powiązać ją z określonym niepustym zbiorem M.

Innymi słowy, interpretacja formuł obliczania predykatów - to uściślenie obszaru nazwowego M i związków między symbolami (stałymi nazwowymi, funkcjonalnymi i predykatywnymi literami), które zawarte są w formułach, z jednej strony, i elementami, funkcjami oraz związkami w M - z drugiej strony.

Rozpatrzmy elementarną formułę:

$$G(f(a,b),\ g(a,b))$$

i jej następującą interpretację:

- M - zbiór liczb rzeczywistych,
- a, b - odpowiednio liczby 2 i 3,
- f - funkcja sumowania ($f(a,b) = a + b$),
- g - funkcja mnożenia ($g(a,b) = a\ b$),
- G - relacja "nie mniejszy".

Przy takiej interpretacji przedstawiona formuła oznacza informację "suma 2 + 3 nie jest mniejsza od wyrażenia 2 3".

Relacja ta nie jest prawdziwa i dlatego:

$$G(f(a,b),\ g(a,b)) = N$$

Jeżeli zmienimy postać interpretacji, podstawiając b = 1 lub b = 2, t $G(f(a,b),\ g(a,b)) = Y$. Istnieje oczywiście wiele innych interpretacji, dla których przytoczona elementarna formuła w pewnych przypadkach ma wartość N, a w innych Y. Nie istnieje jednak żadna interpretacja, dla której relacja ta ma jednoznaczną wartość Y lub N.

Rozważmy inną elementarną formułę:

$$G(f(g(x,x),\ g(y,y)),\ g(a,\ g(x,y)))$$

przy takiej jak wyżej interpretacji.

Formuła ta oznacza informację " $x^2 + y^2 \geq 2\ x\ y$", która jest prawdziwa dla dowolnego x i y z M i przyjmuje wartość Y.

Przy zadanej interpretacji, tj. dla opisania informacji w języku obliczeń predykatów, w zadanym obszarze zmiennych nazwowych często dla oznaczenia predykatowych liter i zmiennych nazwowych lub stałych wykorzystuje się słowa (lub ich skróty), które są nazwami przypisanych im właściwości, związków oraz obiektów.

11. Uwagi końcowe

Pojawienie się wiedzy jako obiektów informacyjnych do przetwarzania w komputerach wyznaczyło procesy przejścia od baz danych do baz wiedzy. Systemy sterowania bazami wiedzy stanowią kolejny etap po systemach sterowania bazami danych. Centralnym problemem podczas tworzenia bazy wiedzy jest wybór sposobu jej reprezentacji - opisu.
Modele obliczeniowo - logiczne reprezentacji wiedzy są przedmiotem zainteresowania autorów i tematem niniejszego artykułu. Autorzy zdają sobie sprawę z konieczności prowadzenia dalszych badań tych zagadnień. W trakcie są próby aplikacji, które w przypadku pomyślnego wyniku powinny doprowadzić do praktycznego wykorzystania zaprezentowanego podejścia.

LITERATURA

[1] Cholewa W., Pedrycz W.: Systemy doradcze. Skrypt Pol. Śl. nr 1447 Gliwice 1987.

[2] Cholewa W., Czogała E.: Podstawy systemów ekspertowych. PAN. prace Inst. Biocybernetyki i Inżynierii Biomedycznej, nr 28, Warszawa 1989.

[3] Grzegorczyk A.: Zarys logiki matematycznej. PWNT. Warszawa 1985.

[4] Mostowski A.W., Pawlak Z.: Logika dla inżynierów. PWN. Warszawa 1970.

[5] Nazaretow W., Kim D.: Techniczeskaja imitacja intelekta. Izd. Wyższaja Szkoła, Moskwa 1986.

[6] Praca zbiorowa pod redakcja Kowalowskiego H.: Sztuczna inteligencja w robotyce i elastycznie automatyzowanej produkcji. Raport z pracy naukowo-badawczej Program RP.I.02. Gliwice 1990.

Recenzent: Doc.dr hab.inż. Jan Kałuski

Wpłynęło do Redakcji do 30.04.1992r.

Abstract

The term "knowledge" in the works on an artifical intelligence is as important as a term "data" in the art of programming.

Methods of "knowledge" representation are considered for creation of the knowledge base. One of the major difficulties in using the knowledge base is determination of models. The aim of this paper is to present theoretical mathematical base for the calculated - logical models of knowledge representation.

In this paper it is shown that it may be very important for artifical intelligence application in practice.