

Metody Inżynierii Wiedzy

Logiczne podstawy inżynierii wiedzy

Antoni Ligęza

Katedra Automatyki
Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Akademia Górniczo-Hutnicza
Kraków

2010

- 1 Logika - wprowadzenie.
- 2 Logika - przegląd logik
- 3 Rachunek zdań
- 4 AI: Systemy logiczne i narzędzia

- 1 Stuart J. Russel, Peter Norvig: *Artificial Intelligence. A Modern Approach*. Second Edition. Prentice Hall, New Jersey, 2003.
- 2 Jay Liebowitz: *The Handbook of Applied Expert Systems*. CRC Press, Boca Raton, 1998.
- 3 Frank van Harmelen, Vladimir Lifschitz, Bruce Porter (Eds.): *Handbook of Knowledge Representation*. Elsevier B.V., Amsterdam, 2008.
- 4 Michael Negnevitsky: *Artificial Intelligence. A Guide to Intelligent Systems*. Addison-Wesley, Pearson Education Limited, Harlow, England, 2002.
- 5 Adrian A. Hopgood: *Intelligent Systems for Engineers and Scientists*. CRC Press, Boca Raton, 2001.
- 6 Joseph C. Giarratano, Gary D. Riley: *Expert Systems. Principles and Programming*. Fourth Edition, Thomson Course Technology, 2005.

- 1 Peter Jackson: *Introduction to Expert Systems*. Addison-Wesley, Harlow, England, 1999.
- 2 Mordechai Ben-Ari: *Mathematical Logic for Computer Science*. Springer-Verlag, London, 2001.
- 3 Antoni Ligęza: *Logical Foundations for Rule-Based Systems*. Springer-Verlag, Berlin, 2006.
- 4 Michael R. Genesereth, Nils J. Nilsson: *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., Los Altos, California, 1987.
- 5 Zbigniew Huzar: *Elementy logiki dla informatyków*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2007.
- 6 Paweł Cichosz: *Systemy uczące się*. WNT, Warszawa, 2000.
- 7 Jan J. Mulawka: *Systemy ekspertowe*. WNT, Warszawa, 1996.

Logika

Nauka zajmująca się badaniem poprawnych rozumowań.

Logika matematyczna

Gałąź matematyki zajmująca się badaniem wnioskowania, jego formalizacją i analizą własności (głębokie podstawy matematyki):

- powstała na przełomie XIX i XX wieku,
- analizuje podstawy matematyki (źródła prawdy; próba formalizacji),
- formalizuje metody rozumowań matematycznych (wnioskowanie, dowodzenie),
- pozwala budować teorie/systemy formalne (język, formuły, reguły wnioskowania, twierdzenia; interpretacje),
- posługuje się aksjomatami, tautologiami, twierdzeniami, hipotezami, faktami, teoriami, etc.
- pozwala dowodzić zdania prawdziwe (twierdzenia) w oparciu o wywód/wnioskowanie z aksjomatów za pomocą reguł wnioskownia.

Paradoks kłamcy

- *ja zawsze kłamię* (Eubulides),
- *Kreteńscy zawsze kłamią* (Epimenides; sam był Kreteńczykiem),
- Kartka: Str.1 *Zdanie na odwrotnej stronie jest prawdziwe*; str.2 *Zdanie na odwrotnej stronie jest fałszywe*.

Matematyka

- Paradoks zbioru wszystkich zbiorów (Cantor, 1899),
- Paradoks Russela (1901): rozważmy zbiór $V = \{X : X \notin X\}$. Czy $V \in V$?
- Paradoks fryzjera: *Pewien fryzjer goli wszystkich mieszkańców miasta, którzy sami się nie golią; co ma zrobić sam ze sobą?*

Przykłady bardziej znanych paradoksów:

<http://pl.wikipedia.org/wiki/Paradoks>

Cechy logiki

- precyzyjnie określona składnia i semantyka,
- logika dostarcza formalnych metod reprezentacji wiedzy,
- wiedza wyspecyfikowana logicznie może być analizowana:
 - wewnętrzna niesprzeczność (spójność),
 - zupełność,
 - minimalna reprezentacja,
 - wynikanie logiczne – logiczna konsekwencja,
 - spełnialność.
- logika dostarcza poprawnych metod wnioskowania,
- logika umożliwia wnioskowanie dedukcyjne, indukcyjne i abdukcyjne,
- aparat logiki jest uniwersalny (ma zastosowanie w wielu dziedzinach, od filozofii i matematyki, poprzez nauki techniczne, do prawa, medycyny i biologii a nawet ekonomii, socjologii i psychologii).

Logiki klasyczne

- rachunek zdań,
- logiki atrybutowe,
- logika Datalogu,
- logika pierwszego rzędu; PROLOG,
- logiki wyższych rzędów.

Logiki nieklasyczne

- logiki wielowartościowe,
- Description Logics (DL),
- logiki niemonotoniczne, logiki domniemań
- logiki modalne,
- logiki temporalne,
- logiki probabilistyczne (Dempster-Shaefer),
- logiki rozmyte.

- prosty język, proste formuły, intuicyjna,
- posługuje się *stwierdzeniami* (fakty, atomy),
- pozwala definiować relacje logiczne pomiędzy stwierdzeniami,
- ograniczona siła ekspresji (brak obiektów, zmiennych, atrybutów, termów, kwantyfikacji),
- problem *niezależności formuł atomicznych*,
- możliwość dowodzenia z wykorzystaniem metody zero-jedynkowej,
- łatwe sprowadzanie do postaci CNF, DNF, NNF; łatwe porównania,
- możliwość zastosowania narzędzi graficznych: tabele Karnaugh'a, diagramy, drzewa, grafy.
- ograniczone możliwości zastosowań praktycznych,
- język modelowy, poglądowy, prezentacyjny.

Reguła odrywania (modus ponens)

$$\frac{\alpha, \alpha \implies \beta}{\beta}$$

Reguła zaprzeczania (modus tollens)

$$\frac{\alpha \implies \beta, \neg\beta}{\neg\alpha}$$

Reguła rezolucji

$$\frac{\phi \vee q, \neg q \vee \psi}{\phi \vee \psi}$$

Reguła rezolucji dualnej

$$\frac{\phi \wedge q, \neg q \wedge \psi}{\phi \wedge \psi}$$

Definicja problemu

- A – pojawił się sygnał do procesu,
- P – sygnał dodany do zbioru sygnałów oczekujących na odebranie przez proces,
- B – sygnał jest zablokowany przez proces,
- D – sygnał został dostarczony do procesu (i odebrany),
- S – stan procesu jest zachowany,
- M – maska sygnałów jest obliczana,
- H – procedura obsługi sygnałów jest wywołana,
- N – procedura obsługi jest wywołana w zwykły sposób,
- R – proces wznowia wykonanie w poprzednim kontekście,
- I – proces musi sam odtworzyć poprzedni kontekst.

Reguły wnioskowania

- $A \longrightarrow P,$
- $P \wedge \neg B \longrightarrow D,$
- $D \longrightarrow S \wedge M \wedge H,$
- $H \wedge N \longrightarrow R,$
- $H \wedge \neg R \longrightarrow I,$

Reguły wnioskowania

$A \rightarrow P,$
 $P \wedge \neg B \rightarrow D,$
 $D \rightarrow S \wedge M \wedge H,$
 $H \wedge N \rightarrow R,$
 $H \wedge \neg R \rightarrow I,$

Fakty – stan systemu

$A, \neg B, \neg R.$

Konkluzje

P
 $P \wedge \neg B$
 D
 $S \wedge M \wedge H$
 $\neg H \vee \neg N$
 $\neg N$
 I

Dowodzenie implikacji (tautologii)

$$(p \implies (q \implies r)) \implies ((p \implies q) \implies (p \implies r))$$

Dowodzenie równoważności (tautologii)

$$((p \implies r) \wedge (q \implies r)) \iff ((p \vee q) \implies r)$$

Dowodzenie wynikania logicznego

$$((p \implies q) \wedge (r \implies s)) \models ((p \vee r) \implies (q \vee s))$$

Wymagania dla języków AI

- odpowiednia siła ekspresji,
- jednoznaczna interpretowalność,
- czytelność dla człowieka i maszyny,
- możliwość automatycznego przetwarzania,
- możliwość automatycznej analizy własności,
- rozszerzalność,
- ...

Koncepcje rozwiązania

- logika, języki logiczne, AL, DL, FOL,
- deklaratywne języki programowania (Prolog, Datalog),
- sieci semantyczne, bazy sieciowe (Freebase),
- specjalizowane języki symboliczne (BPMN, UML),
- ...

Budowa systemu

- fakty (e.g. $p(a, b, c)$),
- reguły (formuły, zależności) (e.g. $p(X) \implies q(X)$; $p(X) \vee q(Y)$),
- reguły wnioskowania (e.g. *modus ponens*, *reguła rezolucji*),
- określamy interpretację I w pewnym uniwersum U (mapowanie symboli języka w stałe, funkcje, i relacje na U).

Podstawy użytkowania

- *fakty* i *formuły* tworzą bazę wiedzy Δ ,
- wychodząc od zadanych *faktów* i *formuł*, w oparciu o przyjęte *reguły wnioskowania* prowadzimy *wywód* nowych faktów ew. formuł,
- jeżeli φ jest wywodliwa z Δ to piszemy $\Delta \vdash \varphi$.
- jeżeli φ jest prawdziwa dla każdej interpretacji dla której Δ jest prawdziwa, to piszemy $\Delta \models \varphi$.

Problemy logiki

- *udowodnienie* (wywód) lub *obalenie* (kontrprzykład) hipotezy,
- budowa systemów niesprzecznych (poprawne reguły dowodzenia):

$$\Delta \vdash \varphi \longrightarrow \Delta \models \varphi,$$

- budowa systemów pełnych (zupełne reguły dowodzenia):

$$\Delta \models \varphi \longrightarrow \Delta \vdash \varphi,$$

- budowa systemów zupełnych:

$$\Delta \models \varphi \text{ or } \Delta \models \neg\varphi,$$

Zastosowania logiki

- modelowanie systemów, dowodzenie ich własności,
- badanie *spełnialności formuł* (problem SAT; poszukiwanie modelu),
- badanie modelu systemu pod kątem własności (ang. *model checking* $M, s \models p$), weryfikacja własności; Constraint Satisfaction (CSP, CLP).

$$h \equiv (p \Leftrightarrow q) \wedge (r \Leftrightarrow s)$$

<i>RuleNo</i>	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>	<i>h</i>
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

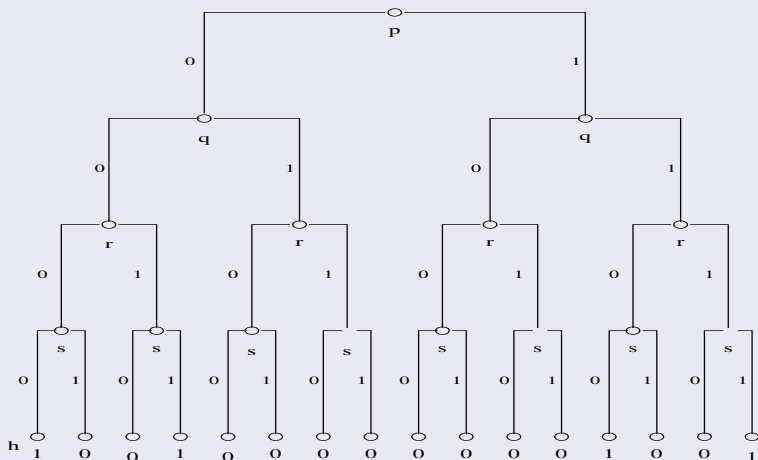
(1)

Binary Decision Tree

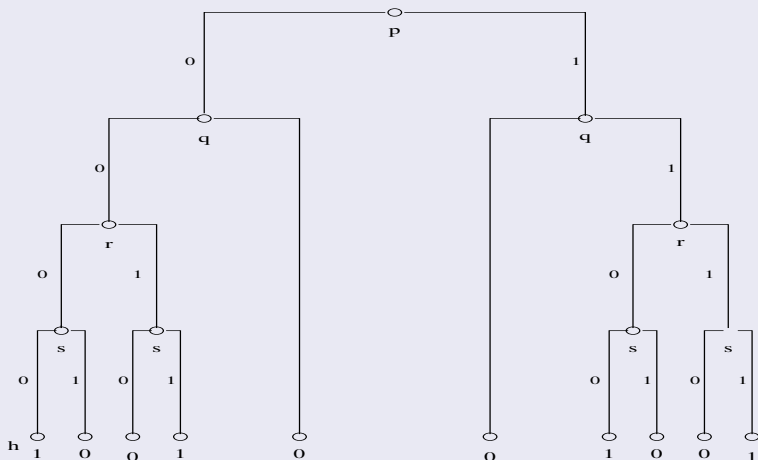
A *Binary Decision Tree* T defined on N , E and P is an acyclic, directed graph satisfying the following properties:

- 1 There is exactly one distinguished node $n_0 \in N$ having no parent node (no entry); it is called the *root node* or the *root* of the tree; we shall write $n_0 = \text{root}(T)$,
- 2 Any other node has exactly one parent node,
- 3 Any node is assigned a single propositional symbol and it has either two child nodes or none:
 - if it has two child nodes, it is a *condition node* and there are two links to child nodes, one referring to the case if the propositional symbol takes *true* and the other one to the case it is *false*;
 - if it has no child nodes, it is a *decision node* or a *leaf node*, and the assigned to it propositional symbol indicates the decision.

Drzewo binarne



Drzewo binarne zredukowane



Notation and rules

$$p \longrightarrow h_0, h_1$$

means a rule of the form:

$$\textit{if } p \textit{ then } h_0 \textit{ else } h_1.$$

Shannon's expansion rule

$$\phi \equiv p \longrightarrow \phi\{p/1\}, \phi\{p/0\},$$

Examples

$$p \wedge q \equiv p \longrightarrow q, 0,$$

$$p \vee q \equiv p \longrightarrow 1, q$$

$$\neg p \equiv p \longrightarrow 0, 1.$$

$$\phi = (p \Leftrightarrow q) \wedge (r \Leftrightarrow s).$$

$$\phi \equiv p \longrightarrow \phi_1, \phi_0 \quad (2)$$

$$\phi_1 \equiv q \longrightarrow \phi_{11}, 0 \quad (3)$$

$$\phi_0 \equiv q \longrightarrow 0, \phi_{00} \quad (4)$$

$$\phi_{11} \equiv r \longrightarrow \phi_{111}, \phi_{110} \quad (5)$$

$$\phi_{00} \equiv r \longrightarrow \phi_{001}, \phi_{000} \quad (6)$$

$$\phi_{111} \equiv s \longrightarrow 1, 0 \quad (7)$$

$$\phi_{110} \equiv s \longrightarrow 0, 1 \quad (8)$$

$$\phi_{001} \equiv s \longrightarrow 1, 0 \quad (9)$$

$$\phi_{000} \equiv s \longrightarrow 0, 1 \quad (10)$$

$$(11)$$

$$\phi \equiv p \longrightarrow \phi_1, \phi_0 \quad (12)$$

$$\phi_1 \equiv q \longrightarrow \phi_{11}, 0 \quad (13)$$

$$\phi_0 \equiv q \longrightarrow 0, \phi_{11} \quad (14)$$

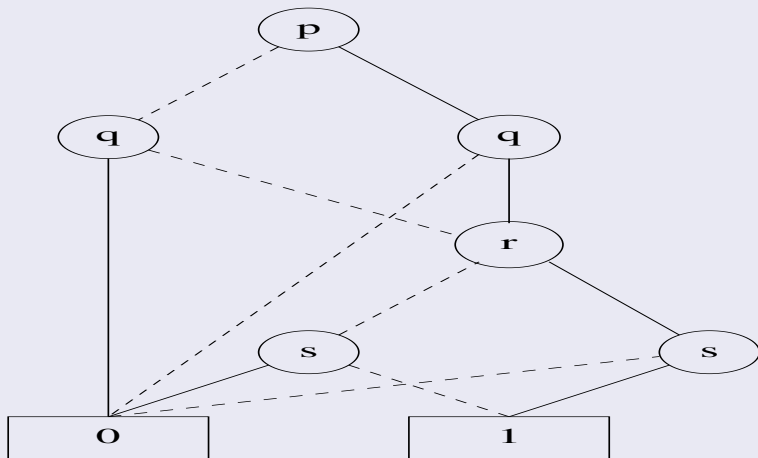
$$\phi_{11} \equiv r \longrightarrow \phi_{111}, \phi_{110} \quad (15)$$

$$\phi_{111} \equiv s \longrightarrow 1, 0 \quad (16)$$

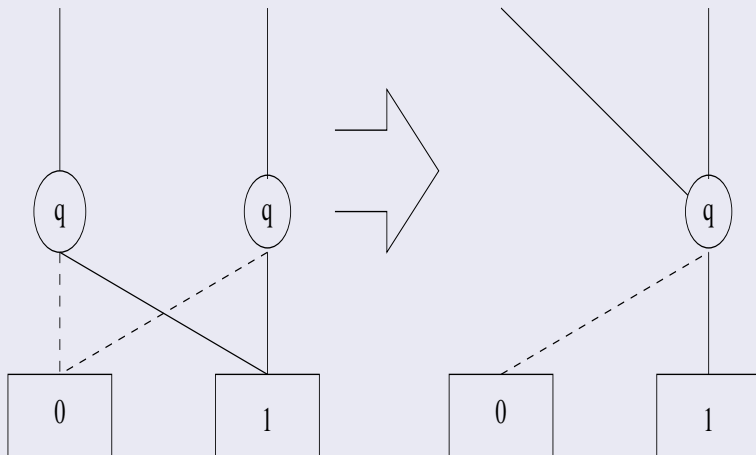
$$\phi_{110} \equiv s \longrightarrow 0, 1 \quad (17)$$

$$(18)$$

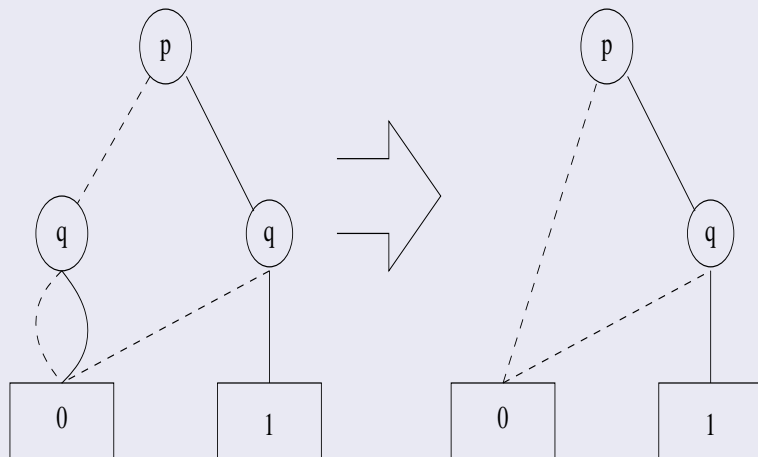
Zredukowany OBDD



Sklejanie identycznych poddrzew



Eliminacja wężła nie wprowadzającego rozgałęzienia



Description Logics

Description Logics is a family of logical languages for knowledge representation and simple inference, based on the syntax of First-Order Predicate Calculus. Its application includes specification of concepts and relations among them as well as their properties. Some distinctive features:

- use of *constants*; constants denote individual objects,
- use of *atomic concepts*; a concept is a set of individual objects,
- use of *atomic roles*; a role is a binary relation between individual objects,
- constructors for *complex objects*,
- constructors for *complex roles*,
- *axioms* defining new concepts,
- *assertions* specifying facts about individuals,
- *concept subsumption*; a concept can be a subconcept of another concept,
- *concept instantiation*; an individual is an instance of a concept.

DL

Symbol	Meaning
\perp	bottom concept (empty set)
\top	universal concept (the set of everything)
$\neg C$	concept complement
$C \sqcap D$	intersection of concepts
$C \sqcup D$	union of concepts
$\exists R.C$	full (limited) existential quantification
$\forall R.C$	value restriction
$C \sqsubseteq D$	subconcept
$C \equiv D$	equivalent concepts
$C(a)$	a is an individual of C
$R(a, b)$	a and b satisfy R

Description Logics and Semantic Web

https://ai.ia.agh.edu.pl/wiki/hekate:dl_intro/