Systemy ekspertowe

Wykład 2
Reprezentacja wiedzy
Rachunek zdań

Joanna Kołodziejczyk

7 maja 2020

Plan wykładu

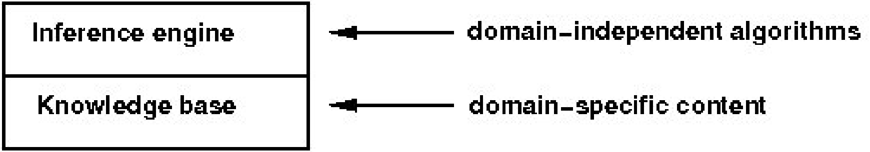
Kodowanie wiedzy

Kodowanie wiedzy oznacza formalny sposób zapisu wiedzy (w postaci symboli), która ma być zgromadzona w systemie.

Do form reprezentacji można zaliczyć:

- postać regułowa (IF - THEN)
- drzewa decyzyjne
- tablice decyzyjne
- sieci semantyczne

Wiedza — jak pokazać w systemie



Baza wiedzy (KB)

zbiór zdań w języku formalnym lub inaczej w języku reprezentacji wiedzy.

Programowanie deklaratywne

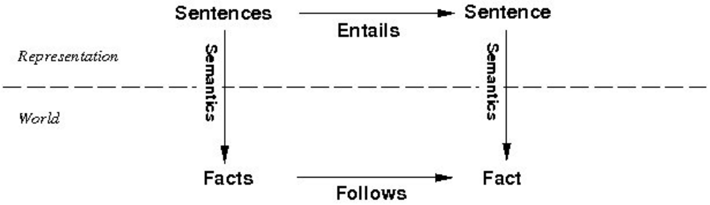
Podejście deklaratywne

podejście ma zapewnić wykonywanie następujących zadań:

- TELL — możliwość poinformowania systemu o nowej wiedzy (wprowadzanie nowych zdań).
- ASK — odpytywanie systemu co jest mu wiadome (odpowiedź powinna wynikać z bazy wiedzy).

ASK i TELL mogą wymagać wnioskowania czyli wyprowadzania nowych zdań na podstawie wcześniej zadanych.

Zależność pomiędzy reprezentacją i rzeczywistością



Representation — formalna reprezentacja wiedzy
World — rzeczywistość
Semnatics — znaczenie
Sentences — zdania reprezentujące wiedzę w jakimś języku formalnym
Facts — rzeczywiste zdarzenia, fakty
Follows — następstwo
Entails — konsekwencja

Logika — podstawy

Logika

Jest formalnym językiem reprezentowania wiedzy o obiektach, z której można wyciągać wnioski (konkluzje) o właściwościach tych obiektów.

Logika elementy:

- Składnia określa budowę zdań w danym języku formalny. Wnioskowanie musi uwzględniać manipulowanie i generowanie symbolami zdań w określonej składni.
- Semantyka określa znaczenie wyrażenia. W logice semantyka definiuje prawdziwość (TRUE) każdego zdania w odniesieniu do rozpatrywanej rzeczywistości.
- System wnioskowania.

Po co stosować język logiki?

- Jako system wnioskowania (*proof system*): Dany jest zbiór faktów (aksjomatów) i zbiór reguł wnioskowania. Celem jest ustalenie, które fakty wynikają z aksjomatów i reguł wnioskowania. W takim spojrzeniu na logikę wykonuje się czysto mechaniczne operacje na symbolach i nie patrzy się na znaczenie zdań, którymi się manipuluje. Nie oznacza to, że dowód nie wymaga kreatywności, ale znaczenie zdania jest w takim wypadku nieistotne.
- Jako teoria modeli (*model theory*): Zdania uzyskują znaczenie, co nazywa się interpretacją. W tym wypadku język logiki jest używany do sformalizowania właściwości struktur i określenia, kiedy zdanie jest prawdziwe. Teoria modeli zmusza do precyzyjnego definiowania pojęcia prawdy. W zależności od interpretacji prawda może mieć zupełnie inne znaczenie.

Poprawność systemu wnioskowania

Cechy systemu wnioskowania

- Niepodważalność (*sound*) każda dowiedziona w nim formuła jest ważna i poprawna.
- Zupełność (*complete*) jeżeli każda ważna i poprawna formuła może być w nim dowiedziona. Systemy zupełne: rachunek zdań, logika predykatów pierwszego rzędu. Systemy niezupełne: logika predykatów drugiego rzędu.

Przykład składni i semantyki

- Arytmetyka to język formalny do reprezentowania zależności i operacji na liczbach.
- Składnia:
  $x + 2 \geq y$ jest poprawnym wyrażeniem a
  $x2 + y >$ nie jest poprawnym wyrażeniem
- Semantyka:
  $x + 2 \geq y$ jest prawdą witw, gdy liczba $x + 2$ jest nie mniejsza niż liczba y
  $x + 2 \geq y$ jest prawdziwe w takiej rzeczywistości, gdzie $x = 7$ i $y = 1$
  $x + 2 \geq y$ jest fałszywe w takiej rzeczywistości, gdzie $x = 0$ i $y = 6$

Logiki

Język reprezentacji	Co opisuje?	Jakie może być przekonanie o faktach?
Rachunek zdań	fakty	true/ false/ nieznany
Rachunek predykatów pierwszego rzędu	fakty , obiekty, relacje	true/ false/ nieznany
Logika temporalna	fakty , obiekty, relacje, czas	true/ false/ nieznany
Teoria prawdopodobieństwa	fakty, reguły	przekonanie w skali [0,1]
Logika rozmyta	fakty, reguły	przekonanie w skali [0,1]

Przykład wnioskowania

Logiczny system wnioskowania: Monty Python — A witch

Składnia rachunku zdań

Rachunek zdań jest najprostszym składniowo systemem logicznym. Pewne założenia przenoszą się jednak na rachunek predykatów pierwszego rzędu.

Alfabet rachunku zdań

1. stałe: *True* i *False*
2. symbole oznaczające zdania (formuły, atomy): P, $Q1$
3. nawiasy okrągłe wokół zdania: $(P \wedge Q)$
4. zdania złożone przez kombinację symboli, stałych z pięcioma symbolami operacji.

Operatory rachunku zdań

1. Negacja $\neg S$. Jeżeli S (positive literal) jest formułą, to $\neg S$ (negative literal) jest formułą.
2. Koniunkcja $\wedge$. Jeżeli S_1 i S_2 to formuły, to $S_1 \wedge S_2$ jest formułą.
3. Dysjunkcja $\vee$. Jeżeli S_1 i S_2 to formuły, to $S_1 \vee S_2$ jest formułą.
4. Implikacja (warunek) $\Rightarrow$. Jeżeli S_1 i S_2 to formuły, to $S_1 \Rightarrow S_2$ jest formułą. Implikacja znana jest też jako reguła czyli zdania typu IF-THEN
5. Równoważność $\Leftrightarrow$. Jeżeli S_1 i S_2 to formuły, to $S_1 \Leftrightarrow S_2$ jest formułą. Czytane: „wtedy i tylko wtedy”.

Semantyka rachunku zdań

- Specyfikuje interpretację każdego symbolu i stałych i określa znaczenie zależności logicznych.
- Znaczenie symboli (ich interpretacja) jest dowolna. np. P może znaczyć „Paryż jest stolicą Francji", czy też „Piotr ma niebieskie oczy"
- P może stać się *True*, jeżeli fakt, o którym mówi zaistniał.
- Zdania złożone mają takie znacznie, które wynika z ich składowych. Można zdania złożone traktować jak funkcje. Gdy podane są wartości wejściowe, to można obliczyć wartość wynikową.

Tablica prawdy

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	true	*false*	*false*	true	true
false	true	true	*false*	true	true	*false*
true	*false*	*false*	*false*	true	*false*	*false*
true	true	*false*	true	true	true	true

Interpretacja implikacji

Przykład

Ojciec obiecuje Jasiowi:

Jeśli $\overbrace{\text{jutro będzie ładna pogoda}}^{p}$, to $\overbrace{\text{pójdziemy na grzyby}}^{q}$.

- Obietnica jest implikacją $p \Rightarrow q$.
- Ojciec nie dotrzyma słowa tylko w jednym przypadku: jeżeli jutro będzie ładna pogoda (tzn. $p = 1$), a nie pójdą z Jasiem na grzyby (tzn. $q = 0$).
- Dlatego przyjmujemy, że implikacja $p \Rightarrow q$ jest fałszywa tylko wtedy, gdy $p = 1$ i $q = 0$.

Walidacja przez tablicę prawdy

Zwalidować zdanie: $((P \vee H) \wedge \neg H) \Rightarrow P$

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
false	*false*	*false*	*false*	true
false	true	true	*false*	true
true	*false*	true	true	true
true	true	true	*false*	true

Zadania

Zapisać w składni rachunku zdań i zwalidować.

1. Jeżeli pan Kowalski sprzedał wrotki i pan Kowalski kupił rower, to nie jest prawdą, że pan Kowalski nie sprzedał wrotek lub nie pan Kowalski nie kupił roweru.
2. Jeżeli Jan Kowalski studiuje, to jest to równoważne, że nie jest prawdą, że Jan Kowalski nie studiuje.
3. Jeżeli Jan Kowalski ma dom i Anna Kowalska ma dom, to wynika z tego, że nie jest prawdą, że Jan Kowalski ma dom lub Anna Kowalska nie ma domu.
4. Jeżeli Nowak był poetą i Kowalski nie był poetą to jest równoważne, że nie jest prawdą, że obaj byli poetami.

Tautologie

$(\alpha \wedge \beta)$	$\Leftrightarrow$	$(\beta \wedge \alpha)$	*przemiennosc* $\wedge$
$(\alpha \vee \beta)$	$\Leftrightarrow$	$(\beta \vee \alpha)$	*przemiennosc* $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\Leftrightarrow$	$(\alpha \wedge (\beta \wedge \gamma))$	*lacznosc* $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\Leftrightarrow$	$(\alpha \vee (\beta \vee \gamma))$	*lacznosc* $\vee$
$\neg(\neg\alpha)$	$\Leftrightarrow$	α	*eliminacja podwojnej negacji*
$(\alpha \Rightarrow \beta)$	$\Leftrightarrow$	$(\neg\beta \Rightarrow \neg\alpha)$	
$(\alpha \Rightarrow \beta)$	$\Leftrightarrow$	$(\neg\alpha \vee \beta)$	*eliminacja implikacji*
$(\alpha \Leftrightarrow \beta)$	$\Leftrightarrow$	$((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	*eliminacja rownowaznosci*
$\neg(\alpha \wedge \beta)$	$\Leftrightarrow$	$(\neg\alpha \vee \neg\beta)$	*De Morgan*
$\neg(\alpha \vee \beta)$	$\Leftrightarrow$	$(\neg\alpha \wedge \neg\beta)$	*De Morgan*
$(\alpha \wedge (\beta \vee \gamma))$	$\Leftrightarrow$	$((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	*rozdzielnosc* $\wedge$ *wzgledem* $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\Leftrightarrow$	$((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	*rozdzielnosc* $\vee$ *wzgledem* $\wedge$

Schematy wnioskowania

Modus ponendo ponens (Modus Ponens)

Zapisujemy: $\dfrac{\begin{array}{c} P \Rightarrow Q \\ P \end{array}}{Q}$ lub jako tautologię: $(P \wedge (P \Rightarrow Q)) \Rightarrow Q)$

- Nazywana regułą dedukcji.
- Jeżeli prawdziwa jest reguła/implikacja (IF THEN) i część przesłankowa reguły P, to możemy wnioskować Q, czyli konsekwencję reguły.
- W sztucznej inteligencji nazywany wnioskowaniem w przód.
- np. $\dfrac{\begin{array}{c} \text{Jeżeli dziś jest niedziela, to jutro jest poniedziałek.} \\ \text{Dziś jest niedziela.} \end{array}}{\text{Jutro jest poniedziałek.}}$

Schematy wnioskowania

Modus ponendo tollens

Zapisujemy: $$\frac{\begin{array}{c}\neg(P \wedge Q)\\ P\end{array}}{\neg Q}$$

lub jako tautologię: $(P \wedge \neg(P \wedge Q)) \Rightarrow \neg Q)$

- Czytamy jako: Albo ... albo
- np. $$\frac{\begin{array}{c}\text{Nie mogę pójść do kina i oglądać telewizji.}\\ \text{Albo pójdę do kina, albo obejrzę telewizję.}\\ \text{Pójdę do kina.}\end{array}}{\text{Nie obejrzę telewizji.}}$$

Schematy wnioskowania

Modus tollendo tollens

Zapisujemy: $$\begin{array}{c} P \Rightarrow Q \\ \neg Q \\ \hline \neg P \end{array}$$

- Nazywany zaprzeczeniem konsekwencji.
- Nie jest możliwe, by przesłanka była prawdziwa i konsekwencja była fałszywa.
- np. $$\begin{array}{c} \text{Jeżeli pies wyczuje obcego, będzie warczał} \\ \text{Pies nie warczał.} \\ \hline \text{Zatem pies nie wyczuł nikogo obcego.} \end{array}$$

Schematy wnioskowania

Modus tollendo ponens (sylogizm dysjunkcyjny/elimienacja dysjunkcji)

Zapisujemy: $$\frac{\begin{array}{c} P \vee Q \\ \neg P \end{array}}{Q}$$

- Wiemy, że przynajmniej jedno ze stwierdzeń jest prawdziwe i wiemy, że inne nie jest prawdziwe, zatem wnioskujemy o prawdziwości drugiego.
- np. $$\frac{\begin{array}{c} \text{Pójdę do kina lub obejrzę telewizję} \\ \text{Nie pójdę do kina.} \end{array}}{\text{Obejrzę telewizję}}$$

Schematy wnioskowania

Sylogizm warunkowy

Zapisujemy: $$\frac{\begin{array}{c} P \Rightarrow Q \\ Q \Rightarrow R \end{array}}{P \Rightarrow R}$$

- np. $$\frac{\begin{array}{c} \text{Jeżeli nie wstanę, to nie pójdę do pracy.} \\ \text{Jeżeli nie pójdę do pracy, nie zarobię.} \end{array}}{\text{Jeżeli nie wstanę, to nie zarobię.}}$$
- Przykład prowadzący do absurdalnych wniosków
$$\frac{\begin{array}{c} \text{Jeżeli Cezar pozostanie w domu, to nie zostanie zabity.} \\ \text{Jeżeli Cezar nie zostanie zabity, to wygłosi przemówienie w senacie.} \end{array}}{\text{Jeżeli Cezar pozostanie w domu, to wygłosi przemówienie w senacie.}}$$
- Błąd wnioskowania wynika z tego, że nie bierze się pod uwagę kontekstowego połączenia przesłanek i konkluzji.

Schematy wnioskowania

Eliminacja koniunkcji

Zapisujemy: $\dfrac{A \wedge B}{B}$ lub $\dfrac{A \wedge B}{A}$

- z iloczynu można wnioskować każdy czynnik
- np. $\dfrac{\text{Bob lubi jabłka i pomarańcze.}}{\text{Bob lubi jabłka.}}$

Schematy wnioskowania

Rezolucja

Zapisujemy: $\dfrac{\begin{array}{c} A \vee \neg B \\ \neg A \vee C \end{array}}{\neg B \vee C}$ z MP $\dfrac{\begin{array}{c} A \Rightarrow B \\ A \end{array}}{B}$ otrzymujemy $\dfrac{\begin{array}{c} \neg A \vee B \\ A \end{array}}{B}$

- Jeżeli A jest prawdziwe, to aby prawdziwa była druga przesłanka, to C musi być prawdziwe. Natomiast jeżeli A jest fałszywe, to aby pierwsza przesłanka była prawdziwa, B musi być prawdziwe.
- np. $\dfrac{\begin{array}{c} \text{Jeżeli ktoś jest Grekiem to jest europejczykiem.} \\ \text{Homer jest Grekiem.} \end{array}}{\text{Homer jest europejczykiem}}$

Koniunkcyjna postać normalna CNF

- Regułę rezolucji stosuje się tylko dla dysjunkcji literałów (zdań). Zatem konieczne będzie posiadanie bazy wiedzy w postaci takich właśnie sum logicznych.
- Każde zdanie w rachunku zdań jest logicznym równoważnikiem koniunkcji dysjunkcji literałów, to znaczy, że dowolne zdanie można na taką formę koniunkcyjną przekształcić.

Koniunkcyjna forma/postać normalna (Conjunctive Normal Form lub CNF)

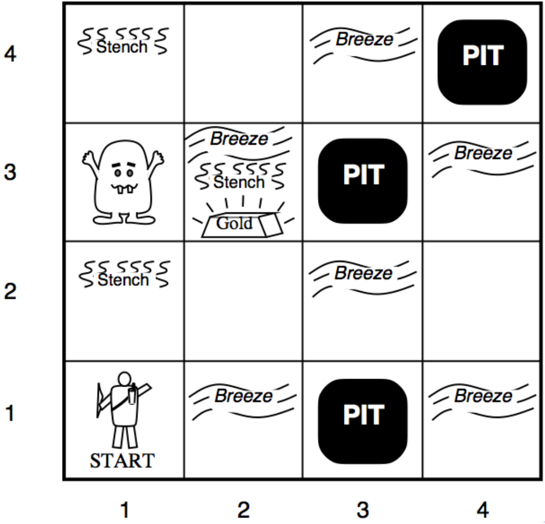
Jest to koniunkcja dysjunkcji literałów czyli iloczyn sum logicznych np.,

$$\overbrace{(A \vee \neg B)}^{klauzula1} \wedge \overbrace{(B \vee \neg C \vee \neg D)}^{klauzula2}$$

Klauzula

Jest to suma literałów. Powyższy przykład zawiera dwie klauzule.

Wumpus by Michael Genesereth



Wumpus - opis

Miara osiągów	złoto +1000; śmierć -1000; -1 za każdy ruch; -10 za użycie strzały
Środowisko	Tablica 4×4 pomieszczeń. Agent zaczyna w polu $[1, 1]$ twarzą skierowaną w prawo. Położenie wumpusa i złota wybierane jest losowo z pominięciem pola startowego. Każde z pól z prawdopodobieństwem 0.2 może być dołkiem
Aktualizatory	Skręć w lewo, skręć w prawo, idź, chwytaj, upuść, strzelaj (do końca wiersza lub kolumny, lub wumpusa), umiera
Czujniki	Pola przylegające do wumpusa cuchną (*stench*). W polach przylegających do dołka czuć powiew (*breeze*). Blask jest w polu gdzie znajduje się złoto (*glitter*). Jeżeli uderzy się w ścianę to jest to sygnalizowane. Zabicie wumpusa jest oznajmiane jego krzykiem. $[Stench, Breez, None, None, None]$

Wumpus World w rachunku zdań

Plansza 4x4.

- Zasady "fizyka"gry:
  - $(B_{x,y} \Rightarrow (P_{x-1,y} \vee P_{x+1,y} \vee P_{x,y-1} \vee P_{x,y+1})$
  - $(S_{x,y} \Rightarrow (W_{x-1,y} \vee W_{x+1,y} \vee W_{x,y-1} \vee W_{x,y+1})$
- Przynajmniej jeden Wumpus na planszy

$$W_{1,1} \vee W_{1,2} \vee W_{1,3} \vee \cdots \vee W_{4,4}$$

- Co najwyżej jeden Wumpus na planszy (dla każdych dwóch sąsiednich pól) $\frac{n(n-1)}{2}$ reguł

$$\neg W_{1,1} \vee \neg W_{1,2}$$

- Nie można umrzeć na starcie

$$\neg P_{1,1}$$

$$\neg W_{1,1}$$

W sumie 155 zdań zawierających 64 symbole.

Algorytm konwersji do koniunkcyjnej postaci normalnej

Zdanie do przekształcenia: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminacja $\Leftrightarrow$, zamień $A \Leftrightarrow B$ na $(A \Rightarrow B) \wedge (B \Rightarrow A)$.

$$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$$

2. Eliminacja $\Rightarrow$, zamień $A \Rightarrow B$ na $\neg A \vee B$.

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$$

3. Przesunięcie $\neg$ do nawiasów stosując prawa de Morgana i podwójną negacje:

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$$

4. Zastosowanie prawa rozdzielności $\vee$ względem $\wedge$:

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

Algorytm rezolucji

Algorytm rezolucji

Nazywany dowodzenia przez sprzeczność (reducio ad absurdum). Aby wykazać, iż A jest spełnione (wynika) z bazy wiedzy KB wykazane zostanie, że $KB \wedge \neg A$ jest niespełnialne, czyli prowadzi do zadnia pustego. Uzyskanie sprzeczności potwierdza postawione założenie.

- Baza wiedzy musi zostać uprzednio przekształcona do postaci CNF.
- Jeżeli dowodzi się złożone zdanie logiczne, to też musi zostać przekształcone na postać CNF.

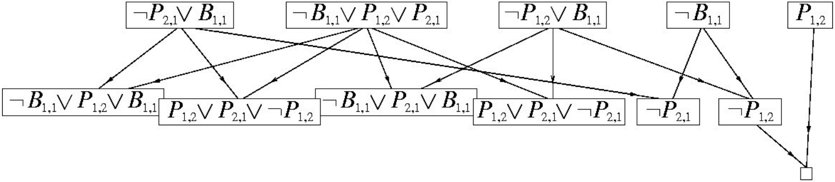
Algorytm rezolucji

1. Zamiana wszystkich zdań w bazie wiedzy i zdania do udowodnienia na postać CNF i utwórzenie z nich jedenego zbiór klauzul roboczych.
2. Zastosuj regułę rezolucji do wszystkich możliwych par klauzul, które zawierają literały komplementarne. W wyniku zastosowania reguły rezolucji powstaną rezolwenty (klauzule bez literałów komplementanych).
3. Jeżeli rezolwenty nie istnieją w zbiorze klauzul roboczych dodaj je do niego.
4. Idź do kroku 2 lub zakończ, gdy uzyskasz zdanie puste (udowodniono hipotezę) lub nie tworzą się żadne nowe klauzule (nie można dowieźć hipotezy z bazy wiedzy).

Przykład dowodzenia z użyciem rezolucji

$KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$ Dowodzimy: $\alpha = \neg P_{1,2}$

Po konwersji $KB \wedge \neg\alpha$ na CNF otrzymujemy pierwszy wiersz na schemacie.



Drugi wiersz powstaje jako rezolwenty z połączeń wszystkich klauzul z pierwszego wiersza. Ostatecznie dwie klauzule zostają połączone prowadząc do klauzuli pustej (sprzeczności) oznaczonej jako mały kwadrat. Zatem dowiedliśmy, że α.

Rachunek zdań: wady i zalety

- Rachunek zdań jest deklaratywny.
- Rachunek zdań dopuszcza częściową/alternatywnę/zanegowaną informacje (w przeciwieństwie do wiekszosci struktur danych i baz danych).
- Rachunek zdań jest zależny od składni: znaczenie $B_{1,1} \wedge P_{1,2}$ wynika ze znaczenia $B_{1,1}$ i $P_{1,2}$.
- Składnia w rachunku zdań jest niezależna od kontekstu (w przeciwieństwie do języka naturalnego).
- Rachunek zdań ma bardzo ograniczoną moc wyrażania (w przeciwieństwie do języka naturalnego), np. nie da się wyrazić zdania „pułapki powodują wiatr w sąsiednich polach" inaczej niż przez napisanie oddzielnego zdania dla każdego pola.

Wykorzystana literatura (do samodzielnego studiowania)

Jean Gallier
Propositional Logic.
http://www.cis.upenn.edu/~cis510/tcl/chap3.pdf

S.J. Russel, P. Norvig
Artificial Intelligence. A modern approach.
Pearson Education wyd. 2, p.111-116