Reprezentacja wiedzy i wnioskowanie

Dr hab. inż. Jerzy Balicki, prof. nadzw.

Plan

1. **Logika pierwszego rzędu**
2. **Elementy rachunku predykatów**
3. **Interpretacja**
4. **Wnioskowanie**
5. **Wynikanie logiczne**

Reprezentacja wiedzy i wnioskowanie

- **Metody reprezentacji wiedzy i automatycznego wnioskowania są ważnym działem informatyki intensywnie obecnie rozwijanym.**
- **Dział ten, łącznie ze zbliżonymi tematycznie metodami przetwarzania języka naturalnego można zaliczyć też do dziedziny sztucznej inteligencji.**
- **Logika pierwszego rzędu zwana też rachunkiem predykatów jest językiem formalnym, w którym można przedstawić opis pewnej rzeczywistości (w postaci faktów i reguł) i przeprowadzać w nim wnioskowanie.**
- **Rachunek predykatow jest rozszerzeniem algebry Boole'a poprzez wprowadzenie zmiennych i kwantyfikatorów. Stworzony został na przełomie XIX i XX wieku – Ch. S. Peirce, G. Frege, B. Russel.**

Wiedza

- „Wszyscy ludzie z natury dążą do poznania, czego dowodem jest ich umiłowanie zmysłów (bo, nawet niezależnie od ich praktycznej użyteczności, miłują je dla nich samych), a zwłaszcza ponad wszystkie inne wzrok.
- Nie tylko bowiem gdy działamy, ale nawet wtedy, gdy nie mamy niczego praktycznego na względzie, stawiamy wzrok ponad wszystkie inne zmysły.
- Przyczyną zaś jest to, że ze wszystkich zmysłów wzrok w najwyższym stopniu umożliwia nam poznanie i ujawnia wiele różnic”

Arystoteles "Metafizyka"

Wiedza

- **W ujęciu „filozoficznym” za wiedzę uznaje się zbiór uzasadnionych przekonań.**
- **W ujeciu „naukowym” za wiedzę uznaje się zbiór uzasadnionych empirycznie lub logicznie/matematycznie stwierdzeń, które można poddawać falsyfikacji i krytyce (K. Popper).**
- **W życiu „potocznym” za wiedzę uznaje się zbiór doświadczeń i przekonań.**
- **Wie ten, kto umie klasyfikować. (Z. Pawlak)**
- **„Knowledge is experience” (A. Einstein)**
- **Wiedza, to warunek podejmowania skutecznych działań**

Dane, informacja, wiedza

- **602 345 123 (ciąg symboli)**
- **Nr telefonu 602 345 123 (dane + interpretacja)**
- **Nr telefonu do Ewy 602 345 123 (informacje powiązane relacjami)**

Reprezentacja wiedzy

- **Główną siłą sprawczą wyznaczającą zakres i kierunek prac nad reprezentowaniem wiedzy jest to, do czego owa reprezentacja ma być stosowana oraz to, w jaki sposób wiedza będzie pozyskiwana.**
- **Nie istnieje zatem jedna, akceptowana przez wszystkich definicja terminu *reprezentacja wiedzy*.**
- **„Reprezentowanie wiedzy polega na tworzeniu opisów świata lub jego stanów” R. Brachman, H. Levesque, 1985**

Reprezentacja wiedzy

- **Przez reprezentacje wiedzy rozumie się tu sposób w jaki wiedza o świecie jest przedstawiana wraz z metodami przetwarzania, a zwłaszcza wnioskowania (inferencji).**

Reprezentacja wiedzy

- **Przez reprezentacje wiedzy rozumie się tu sposób w jaki wiedza o świecie jest przedstawiana wraz z metodami przetwarzania, a zwłaszcza wnioskowania (inferencji).**

Metody reprezentacja wiedzy

- **język naturalny,**
- **metody stosowane w obszarze baz danych, np. UML**
- **logika matematyczna (klasyczna, niestandardowa),**
- **reguły produkcji (*production rules),***
- **sieci semantyczne (*semantic networks),***
- **grafy koncepcji *(concept graphs),***
- **ontologie,**
- **ramy, scenariusze (*frames, scripts),***
- **zbiory przybliżone (*rough sets),***
- **XML**
- **sieci neuronowe (neural nets),**
- **algorytmy genetyczne (*genetic algorithms) …***

Logika pierwszego rzędu

- **W latach 30. XX w. badano zagadnienia pełności i rozstrzygalności systemów wnioskowania (Gödel, Herbrand, Skolem, Tarski).**
- **W 1965 A. Robinson podał algorytm dowodu oparty o rezolucję.**
- **Duży wkład w rozwoj logiki matematycznej wnieśli też uczeni polscy (Chwistek, Jaśkowski, Kuratowski, Leśniewski, Łukasiewicz, Mostowski, Rasiowa, Słupecki, Sobociński)**

W aksjomatycznej teorii matematycznej zawierającej pojęcie liczb naturalnych da się sformułować takie zdanie, którego w ramach tej teorii nie da się ani udowodnić, ani obalić. (Twr. o niezupełności Kurta Gödla, 1931)

Predykat

- **funktor zdaniotwórczy** od co najmniej jednego argumentu nazwowego;
- **funkcję zdaniową** argumentów nazwowych;
- **wyrażenie** opisujące pewne **własności** lub **relacje**.

Twr. Gödla zakończyło definitywnie setkę lat prób zaksjomatyzowania całej matematyki, z twierdzenia wynika, że jest to zadanie niewykonalne. Ponadto matematyka nie jest i nie może być nauką zamkniętą i zakończoną.
Nie istnieje program komputerowy, który zdoła rozstrzygnąć wszystkie problemy matematyczne. Istnieją konkretne problemy, których nie da się rozwiązać na żadnym komputerze.

Wyrażenia poprawnie zbudowane

- Prawidłowe wyrażenia rachunku predykatów nazywane są **wyrażeniami poprawnie zbudowanymi** (**wpz**).
- Poprawne jest wyrażenie

$$p(f(X), Y))$$

gdzie *f* nazywa się **funktorem** lub **literą funkcyjną**.

- Wyrażenie otrzymane przez **kwantyfikację** wpz po pewnej zmiennej jest także wpz.
- Jeżeli **zmienna** w wpz została skwantyfikowana, to nazywa się ją **zmienną związaną**, a w przeciwnym wypadku **zmienną swobodną**.
- Wpz, w których **nie ma zmiennych swobodnych**, nie są już predykatami, a reprezentują pewne **zdania twierdzące**. Takie wyrażenia nazywają się **zdaniami**.

Rachunek predykatów

- **W rachunku predykatów często wykorzystuje się poniższe formuły**

$$\neg(\exists X)P(X) \equiv (\forall X)[\neg P(X)]$$

$$\neg(\forall X)P(X) \equiv (\exists X)[\neg P(X)]$$

- Terminologia:

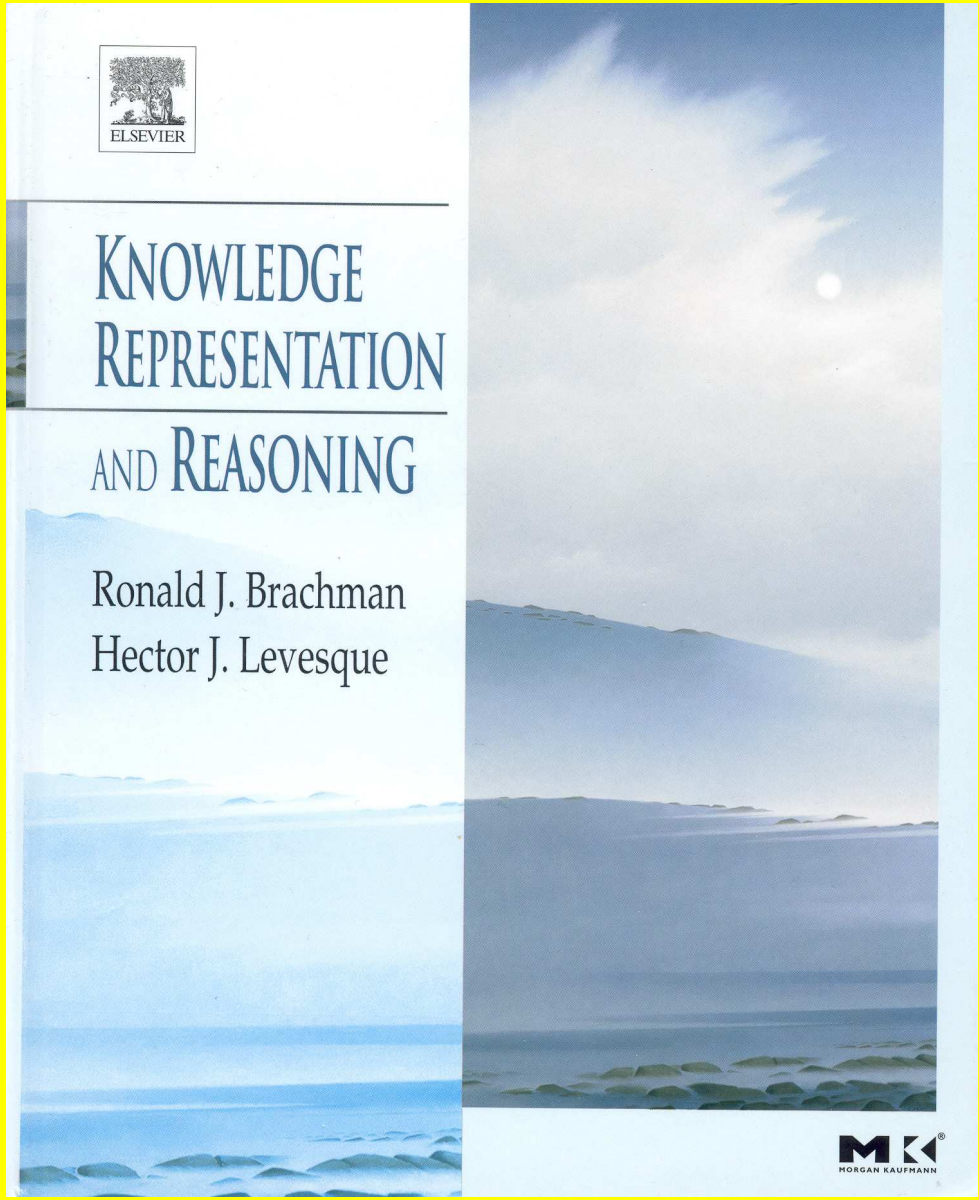
atom – predykat

literał – atom lub atom zanegowany

term – argumenty atomu

Reprezentacja wiedzy i wnioskowanie

1. Wprowadzenie do logiki pierwszego rzędu (kwantyfikowane tylko zmienne)



Język logiki pierwszego rzędu

- predykaty mogą opisywać jakąś relację, np:

$$kocha(X, Y)$$

lub własność elementu.

Predykat przyjmuje wartość:
Prawda (1), dla obiektów X i Y spełniających relację.
Fałsz (0), w przeciwnym razie,

Argumentami predykatu mogą być zmienne, stałe i funkcje np.:

$$p(X, Y, a, gdynia, f(X))$$

Język logiki pierwszego rzędu

- operatory logiczne (boolowskie) - funktory, np.:

$$\vee, \wedge, \rightarrow, \oplus, \equiv, \neg,$$

- kwantyfikatory (kolejność może mieć znaczenie)

a) uniwersalny (duży)

$(\forall X)$ – dla każdego X

b) egzystencjalny (mały)

$(\exists X)$ – istnieje X

Język logiki pierwszego rzędu

Wyrażenie predykatowe, w którym wszystkie zmienne objetę są działaniem kwantyfikatora nazywa się zamkniętym i wyraża pewne zdanie twierdzące, które może być prawdziwe lub fałszywe.

Wyrażenie predykatowe, w którym występują zmienne nieobjęte działaniem kwantyfikatorów i nie reprezentuje zdania twierdzącego, ma własności predykatu - może przyjmować w zależności od podstawienia obiektów za zmienne wartość: Prawda lub Fałsz.

Prawidłowe wyrażenia rachunku predykatów nazywane są wyrażeniami poprawnie zbudowanymi (wpz).

Predykat jako funktor zdaniotwórczy

Wyrażenie:

$$pod(kot, stół)$$

może reprezentować zdanie: „Kot znajduje się pod stołem”

Predykat dwuargumentowy.

Predykat jako funktor zdaniotwórczy

od co najmniej jednego argumentu nazwowego.

Przykłady

Zdanie: „Istnieje osoba, która napisała ‚Pana Cogito' ”
(Zbigniew Herbert, 1974)
(Cogito ergo sum - Kartezjusz)

może być reprezentowane przez wyrażenie:

$$(\exists X) napisal(X, ,,Pan\ Cogito'')$$

Przykłady

Wyrażenie:

$$\neg(\exists X)\,[\mathit{mrówka}(X) \wedge \mathit{inteligentny}(X)]$$

może reprezentować zdanie:

„Nie ma inteligentnych mrówek”.

Human: Tell me about your programming.
ALICE: Some people call my technology "Case Based Reasoning" or CBR.
Human: So, tell me about CBR.
ALICE: Cased-Based Reasoning is another name for "nearest-neighbor" classification. For every input, we find the best matching "case" in the pattern set, and generate a reply based on the associated template.

Przykłady

Wyrażenia:

$$(\forall X \forall Y) dziecko(X, Y) \rightarrow potomek(X, Y)$$
$$(\forall X \forall Y \forall Z) dziecko(Z, Y) \wedge potomek(X, Z)$$
$$\rightarrow potomek(X, Y)$$

przedstawiają zdanie (regułę):
„Dziecko (jakiejś osoby) jest (jej) potomkiem
i potomek dziecka (jakiejś osoby)
jest potomkiem (tej osoby)”
co jest rekurencyjną definicją bycia potomkiem.

Przykłady

Wyrażenia

$$(\forall X)(\forall Y) rodzic(X, Y) \rightarrow przodek(X, Y)$$

$$(\forall X)(\forall Y)(\forall Z) rodzic(X, Z) \wedge przodek(Z, Y) \rightarrow$$
$$\rightarrow przodek(X, Y)$$

przedstawiają zdanie (regułę):
„Jeżeli X jest rodzicem Y,
to jest jego przodkiem lub
jeżeli X jest rodzicem (jakiegoś) Z i Z jest przodkiem Y,
to X jest przodkiem Y”,
Rodzic (jakiejś osoby) jest (jej) przodkiem
i rodzic przodka (kogoś) jest przodkiem (tego kogoś),
co jest rekurencyjną definicją bycia przodkiem.

2. Interpretacja

Przez interpretację rozumiemy dowolny system składający się z niepustego zbioru D nazywanego dziedziną interpretacji, a także z zależności:

- każdemu predykatowi: P^n, n-wymiarową relację w D,
- każdej literze funkcyjnej: f^n, n-wymiarową funkcję w D
- każdej stałej: a_j element z D.

Interpretacja definiuje semantykę języka predykatów.

Interpretacja – przykład

Rozpatrzmy wpz:

$$p(X, Y)$$

Przy interpretacji:

D – zbiór liczb całkowitych dodatnich
$p(X, Y)$ wyraża relację $X \leq Y$

powyższe wpz przedstawia relację:

$$X \leq Y$$

w dziedzinie liczb całkowitych dodatnich.

Interpretacja – przykład

Rozpatrzmy wpz:

$$(\forall Y)p(X,Y)$$

Przy interpretacji:

D – zbiór liczb całkowitych dodatnich
$p(X,Y)$ wyraża relację $X \leq Y$

powyższe wpz przedstawia relację:
X jest mniejsze lub równe od każdej liczby całkowitej dodatniej (co jest spełnione dla $X = 1$)

Interpretacja – przykład

Rozpatrzmy wpz:

$$(\exists X \forall Y) p(X, Y)$$

Przy interpretacji:

D – zbiór liczb całkowitych dodatnich
$p(X, Y)$ wyraża relację $X \leq Y$

powyższe wpz przedstawia zdanie (twierdzenie):

Istnieje liczba, która jest mniejsza od każdej liczby całkowitej dodatniej (istnieje najmniejsza liczba całkowita dodatnia).
Prawda jedynie dla l.calk. dodatnich.

3. Wnioskowanie

Wnioskowaniem nazywamy proces stosujący *reguły wnioskowania* dla uzyskania wpz z zadanego zbioru wpz.

Wpz otrzymane drogą wnioskowania nazywa się *wnioskiem*, a ciąg zastosowanych *reguł wnioskowania* tworzy dowód.

Wyrażenie

$$S \vdash W$$

oznacza, że wpz W jest wnioskiem ze zbioru $S = \{S_1, S_2, \ldots, S_n\}$.

Reguły wnioskowania

Wnioskowanie jest procesem wielokrokowym, gdzie w każdym kroku stosowana jest odpowiednio dobrana *reguła wnioskowania*.

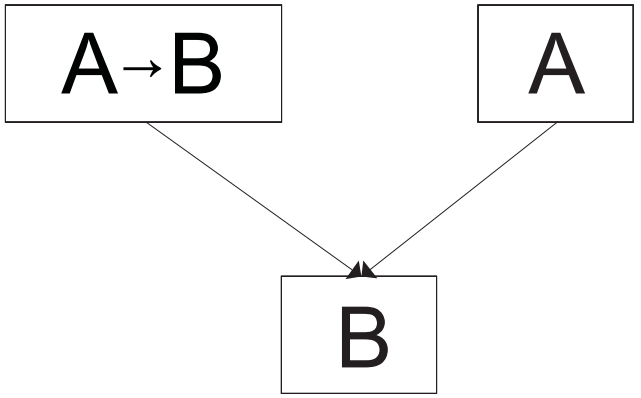
Reguła wnioskowania składa się ze:
a) zbioru wyrażeń nazywanych *warunkami*
b) zbioru wyrażeń nazywanych *konkluzjami*.

W ten sposób stosując reguły wnioskowania
z wpz i zbiorów wpz otrzymujemy
nowe wpz, aż do uzyskania wymaganego wyrażenia.

Reguły wnioskowania

reguła odrywania (modus ponens)

ze zdania (reguły) *jeżeli* A *to* B, i zdania (faktu) A,
wynika konkluzja (fakt) B.



If today is Wednesday, then I will go to AI lecture.
Today is Wednesday.
Therefore, I will go to AI lecture.

Modus ponens jest nazywany wnioskowanie w przód.

uniwersalna specjalizacja :

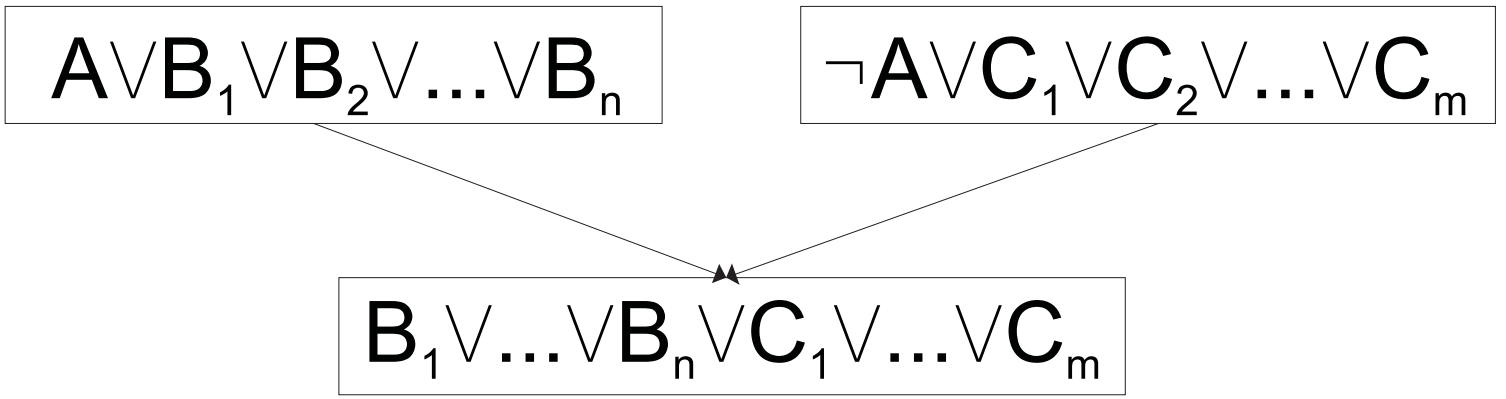
wytwarza wpz $W(a)$ z wpz $(\forall X)W(X)$.

reguła rezolucji ze zdania *A lub B*

i zdania *nie A lub C* wynika zdanie *B lub C*.

Reguła rezolucji zawiera, jako szczególne przypadki, inne reguły wnioskowania.

Rezolucja to metoda automatycznego dowodzenia twierdzeń oparta na generowaniu nowych klauzul aż dojdzie się do sprzeczności. W ten sposób można udowodnić, że dane twierdzenie nie jest spełnialne, lub też, co jest równoważne, że jego zaprzeczenie jest tautologią.



Klauzula

Klauzula (ang. clause) to zbiór literałów, który jest prawdziwy wtedy i tylko wtedy, gdy ich alternatywa jest prawdziwa. Klauzula pusta jest zawsze fałszywa.

Klauzulę zapisujemy jako wpz w postaci sumy logicznej literałów:

$$L_1 \vee L_2 \vee \ldots \vee L_n$$

gdzie L_i, $i = 1, 2, \ldots, n$ są literałami. Na przykład:

$$P(X) \vee \neg P(Y) \vee Q(X, Y, Z)$$

Klauzule Horna

Klauzule, które zawierają tylko jeden nie zanegowany atom nazywają się *klauzulami Horna*

$$A_1 \vee \neg A_2 \vee \ldots \vee \neg A_n$$

gdzie A_i, $i = 1, 2, \ldots, n$ są atomami.

Przykładami takich klauzul są {p,¬r,¬q} i {¬r,¬q}.

W Prologu p :- r, q odpowiada p ← r and q

Klauzule Horna można zapisać w postaci implikacji:

$$A_n \wedge A_{n-1} \wedge \ldots \wedge A_2 \rightarrow A_1$$

Strategie sterowania wnioskowaniem

Przy stosowaniu reguły odrywania możliwe są dwie podstawowe strategie sterowania wnioskowaniem:

a) w przód (ang. forward chaining) i
b) w tył (ang. backward chaining).

Strategie sterowania wnioskowaniem

Wnioskowanie w tył. Należy wykazać A.

Baza wiedzy				Wnioskowanie	
(1)	A	$\leftarrow$	$B \wedge C$	1	Aby wykazać A należy wykazać B i C
(2)	A	$\leftarrow$	$C \wedge D$	2	Aby wykazać B należy wykazać F i E
(3)	B	$\leftarrow$	$F \wedge E$	3	Aby wykazać F należy wykazać R
(4)	C	$\leftarrow$	$R \wedge S$	4	R jest faktem, więc R jest wykazane
(5)	F	$\leftarrow$	R	5	R jest wykazane, więc F jest wykazane
(6)	D			6	E jest faktem, więc E jest wykazane
(7)	E			7	B jest wykazane
(8)	R			8	Aby wykazać C należy wykazać R i S
(9)	S			9	R jest faktem, więc R jest wykazane
				10	S jest faktem, więc S jest wykazane
				11	C jest wykazane
				12	A jest wykazane

Strategie sterowania wnioskowaniem

Wnioskowanie w przód. Należy wykazać A.

Baza wiedzy	Wnioskowanie
(1) $A \leftarrow B \wedge C$	(10) (8)(5) F
(2) $A \leftarrow C \wedge D$	(11) (3)(7)(10) B
(3) $B \leftarrow F \wedge E$	(12) (4)(8)(9) C
(4) $C \leftarrow R \wedge S$	(13) (1)(11)(12) A
(5) $F \leftarrow R$	
(6) D	
(7) E	
(8) R	
(9) S	

Wynikanie logiczne

Dany jest zbiór wpz:

$$S_1, S_2, \ldots, S_n$$

Jeżeli dana interpretacja powoduje, że każde wpz w tym zbiorze jest prawdziwe (ma wartość T), wtedy mówimy, że interpretacja *spełnia* zbiór wpz (jest jego *modelem*).

Wpz W *wynika logicznie* ze zbioru wpz S jeżeli każda interpretacja spełniająca S także spełnia W, co oznaczamy $S \models W$.

Wynikanie logiczne

wynikanie logiczne (konsekwencja)	$S \models W$
wnioskowanie logiczne (wniosek)	$S \vdash W$

Twierdzenie Jeżeli S jest zbiorem wpz, zaś W jest wpz, to: $S \models W$ wtedy i tylko wtedy, gdy $S \vdash W$

Dowodzenie przez sprzeczność

Dany jest zbiór wpz:

$$S = \{S_1, S_2, \ldots, S_n\}$$

Należy wykazać, że wpz W jest wnioskiem ze zbioru S.

W metodzie dowodzenia przez wykazanie sprzeczności neguje się W i dołącza $\neg W$ do zbioru S. Jeżeli W wynika logicznie z S, to zbiór S i $\neg W$ nie jest spełniony przy żadnej interpretacji.

Dowodzenie przez sprzeczność

Aby wykazać, że zbiór wpz nie jest spełniony przy żadnej interpretacji, wystarczy pokazać, że jest sprzeczny, to znaczy że można z niego wyprowadzić zarówno pewne wpz P jak i wpz $\neg P$, z których wynika klauzula pusta *NIL* oznaczająca sprzeczność.

Aby to wykaza**ć**, należy przekształcić zbiór wpz do postaci klauzul, a następnie stosować wielokrotnie regułe rezolucji aż do uzyskania klauzuli pustej *NIL*.

wpz $\rightarrow$ klauzule

Kolejne kroki przekształcania pokazane są na poniższym przykładzie

$$(\forall X)\,[p(X) \rightarrow [(\forall Y)\,[p(Y) \rightarrow p(f(X,Y))]$$
$$\wedge\neg(\forall Y)\,[q(X,Y) \rightarrow p(Y)]]]$$

wpz $\rightarrow$ klauzule

$$(\forall X)\,[p(X) \rightarrow [(\forall Y)\,[p(Y) \rightarrow p(f(X,Y))]$$
$$\wedge\neg(\forall Y)\,[q(X,Y) \rightarrow p(Y)]]]$$

1. Eliminacja symboli implikacji

Korzystamy ze wzoru

$$A \rightarrow B \equiv \neg A \vee B$$

Otrzymujemy

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge\neg(\forall Y)\,[\neg q(X,Y) \vee p(Y)]]]$$

wpz $\rightarrow$ klauzule

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge \neg(\forall Y)\,[\neg q(X,Y) \vee p(Y)]]]$$

2. Redukcja zakresu działania symbolu negacji

Korzystamy tu z wzorów

$$\begin{aligned}
\neg(A \wedge B) &\equiv \neg A \vee \neg B \\
\neg(A \vee B) &\equiv \neg A \wedge \neg B \\
\neg\neg A &\equiv A \\
\neg(\forall X)A(X) &\equiv (\exists X)\neg A(X) \\
\neg(\exists X)A(X) &\equiv (\forall X)\neg A(X)
\end{aligned}$$

wpz $\rightarrow$ klauzule

Mamy

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge \neg (\forall Y)\,[\neg q(X,Y) \vee p(Y)]]]$$

Otrzymujemy

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge (\exists Y)\,[q(X,Y) \wedge \neg p(Y)]]]$$

wpz $\rightarrow$ klauzule

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge(\exists Y)\,[q(X,Y) \wedge \neg p(Y)]]]$$

3. Standaryzacja zmiennych

Każdy kwantyfikator działa na zmienną o innej nazwie, np:

$$(\forall X)[p(X) \rightarrow (\exists X)q(X)] \equiv (\forall X)[p(X) \rightarrow (\exists Y)q(Y)]$$

Otrzymujemy

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge(\exists Z)\,[q(X,Z) \wedge \neg p(Z)]]]$$

wpz $\rightarrow$ klauzule

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge(\exists Z)\,[q(X,Z) \wedge \neg p(Z)]]]$$

4. Eliminacja kwantyfikatorów szczegółowych

Korzystamy tu ze wzorów:

$$\begin{aligned}(\exists X)p(X) &\equiv p(a)\\ (\exists X\forall Y)p(X,Y) &\equiv (\forall Y)p(a,Y)\\ (\forall Y\exists X)p(X,Y) &\equiv (\forall Y)p(g(Y),Y)\end{aligned}$$

gdzie $g(\cdot)$ – funkcja Skolema.

wpz $\rightarrow$ klauzule

Mamy

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge (\exists Z)\,[q(X,Z) \wedge \neg p(Z)]]]$$

Otrzymujemy

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge\,[q(X,g(X)) \wedge \neg p(g(X))]]]$$

wpz $\rightarrow$ klauzule

$$(\forall X)\,[\neg p(X) \vee [(\forall Y)\,[\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge\,[q(X,g(X)) \wedge \neg p(g(X))]]]$$

5. Przesunięcie kwantyfikatorów uniwersalnych na początek wyrażenia

Otrzymujemy

$$(\forall X \forall Y)\,[\neg p(X) \vee [\neg p(Y) \vee p(f(X,Y))]$$
$$\wedge\,[q(X,g(X)) \wedge \neg p(g(X))]]$$

wpz $\rightarrow$ klauzule

$$(\forall X \forall Y)\,[\neg p(X) \vee [\neg p(Y) \vee p(f(X,Y))]$$

$$\wedge\,[q(X, g(X)) \wedge \neg p(g(X))]]$$

6. Przedstawienie wyrażenia w normalnej postaci iloczynowej

Korzystamy tu ze wzoru

$$A \vee B_1 \wedge B_2 \wedge \ldots \wedge B_n \equiv (A \vee B_1) \wedge (A \vee B_2) \wedge \ldots \wedge (A \vee B_n)$$

Otrzymujemy

$$(\forall X \forall Y)\,[[\neg p(X) \vee \neg p(Y) \vee p(f(X,Y))]$$

$$\wedge\,[\neg p(X) \vee q(X, g(X))] \wedge [\neg p(X) \vee \neg p(g(X))]]$$

wpz $\rightarrow$ klauzule

$$(\forall X \forall Y)\,[[\neg p(X) \vee \neg p(Y) \vee p(f(X,Y))]$$

$$\wedge\,[\neg p(X) \vee q(X, g(X))] \wedge [\neg p(X) \vee \neg p(g(X))]]$$

7. Eliminacja kwantyfikatorów uniwersalnych

Wszystkie zmienne w wpz są kwantyfikowane uniwersalnie. Ponieważ kolejność kwantyfikatorów uniwersalnych nie ma znaczenia, to nie piszemy ich jawnie. Otrzymujemy

$$[\neg p(X) \vee \neg p(Y) \vee p(f(X,Y))]$$

$$\wedge\,[\neg p(X) \vee q(X, g(X))] \wedge [\neg p(X) \vee \neg p(g(X))]$$

wpz $\rightarrow$ klauzule

$$[\neg p(X) \vee \neg p(Y) \vee p(f(X,Y))]$$

$$\wedge [\neg p(X) \vee q(X, g(X))] \wedge [\neg p(X) \vee \neg p(g(X))]$$

6. Eliminacja symboli koniunkcji
Wypisujemy wyrażenia rozdzielone symbolem koniunkcji – otrzymujemy wyrażenie początkowe doprowadzone do postaci klauzul.

$$\neg p(X) \vee \neg p(Y) \vee p(f(X,Y))$$

$$\neg p(X) \vee q(X, g(X))$$

$$\neg p(X) \vee \neg p(g(X))$$

Unifikacja

to poszukiwanie podstawień termów za zmienne.

W logice proces ujednolicania, w wyniku którego zakresy pojęciowe lub znaczenia niezwiązane ze sobą lub w jakiś sposób niezgodne,
nabywają zgodności i stają się częścią większej całości.

Unifikacja

Przykład Rozpatrzmy proces unifikacji dwóch literałów

$$P(a, X, f(g(Y))), \qquad P(Z, f(Z), f(U))$$

Przeglądając wyrażenia od strony lewej do prawej, poszukując termów lub części termów, które są różne i znajdując podstawienia S zrównujące je, kolejno otrzymujemy

Unifikacja

$$P(a, X, f(g(Y))), \qquad P(Z, f(Z), f(U))$$

$$S = \{a/Z\}$$

$$P(a, X, f(g(Y))), \qquad P(a, f(a), f(U))$$

$$S = \{a/Z, f(a)/X\}$$

$$P(a, f(a), f(g(Y))), \qquad P(a, f(a), f(U))$$

$$S = \{a/Z, f(a)/X, g(Y)/U\}$$

$$P(a, f(a), f(g(Y))), \qquad P(a, f(a), f(g(Y)))$$

Unifikacja

Przykład

$$P(f(X, g(a, Y)), g(a, Y)), \quad P(f(X, Z), Z)$$

$$S = \{g(A, Y)/Z\}$$

$$P(f(X, g(a, Y)), g(a, Y)), \quad P(f(X, g(a, Y)), g(a, Y))$$

Unifikacja

Przykład

$$P(X, f(X, f(X, X))), \quad P(f(Y, Y), Y)$$

$$S = \{f(Y, Y)/X\}$$

$$P(f(Y, Y), f(f(Y, Y), f(f(Y, Y), f(Y, Y)))),$$

$$P(f(Y, Y), Y)$$

W następnym podstawieniu pojawi się

$$S = \{\ldots, (\text{ funkcja zmiennej } Y)/Y\}$$

co prowadzi do nieskończonej rekurencji i w konsekwencji powoduje, że wyrażenia nie są unifikowalne.

Rezolucja dwóch klauzul

Dane są dwie klauzule

$$L = \bigvee_i l_i, \qquad M = \bigvee_i m_i$$

gdzie l_i i m_i są literałami. Niech

$$l_j \subseteq \{l_i\}, \qquad m_k \subseteq \{m_i\}$$

Jeżeli istnieje unifikator S dla l_j i $\neg m_k$, to klauzule L i M mają rezolwentę (wynik wnioskowania m. rezolucji)

$$\left\{\bigvee_{i \neq j} l_i\right\}_S \vee \left\{\bigvee_{i \neq k} m_i\right\}_S$$

Rezolucja dwóch klauzul

Przykład Stosując podstawienie

$$S = \{X/Z, a/X\}$$

dwie klauzule

$$P(X, f(a)) \vee P(X, f(Y)) \vee \neg Q(Y), \quad \neg P(Z, f(Z)) \vee R(Z)$$

zostają przekształcone do postaci

$$P(a, f(a)) \vee P(a, f(Y)) \vee \neg Q(Y), \quad \neg P(a, f(a)) \vee R(a)$$

i ich rezolwenta wynosi

$$P(a, f(Y)) \vee \neg Q(Y) \vee R(a),$$

Przykład wnioskowania

Dany jest zbiór formuł rachunku predykatów (zbiór wpz)

1. $(\forall X)[pk(X) \rightarrow p(X)]$
2. $(\forall X)[d(X) \rightarrow \neg p(X)]$
3. $(\exists X)[d(X) \wedge i(X)]$

Należy udowodnić, że:

4. $(\exists X)[i(X) \wedge \neg pk(X)]$

Przykład wnioskowania

Wyrażenia 1, 2, 3 wraz z zanegowaną tezą 4 po przekształceniu do postaci klauzul:

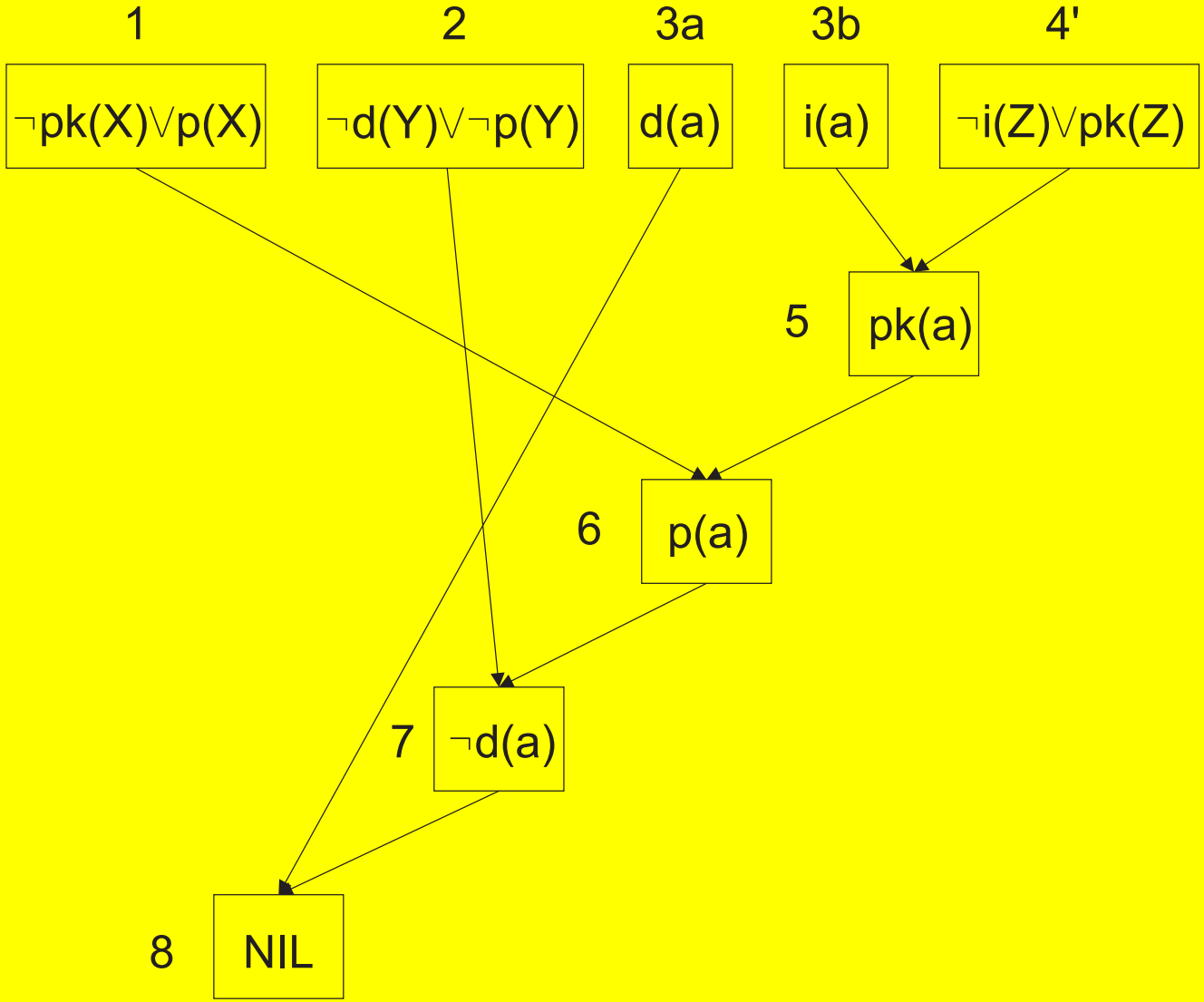
wyrażenie	klauzula
$(\forall X)[pk(X) \rightarrow p(X)]$	$\neg pk(X) \vee p(X)$
$(\forall X)[d(X) \rightarrow \neg p(X)]$	$\neg d(Y) \vee \neg p(Y)$
$(\exists X)[d(X) \wedge i(X)]$	$d(a)$ i $i(a)$
$\neg(\exists X)[i(X) \wedge \neg pk(X)]$	$\neg i(Z) \vee pk(Z)$

Przykład wnioskowania

Stosując rezolucję do wybranych par klauzul otrzymujemy kolejno:

5. $pk(a)$ rezolwenta 3b i 4'
6. $p(a)$ rezolwenta 5 i 1
7. $\neg d(a)$ rezolwenta 6 i 2
8. *NIL* rezolwenta 7 i 3a.

Przykład wnioskowania



Przykład wnioskowania

Zauważmy, że np. nadając predykatom $pk(X)$, $p(X)$, $d(X)$, $i(X)$ interpretację: programujący komputery, programista, delfin i inteligentny, udowodniliśmy powyżej, że ze zdań:

> programujący komputery jest programistą
> delfiny nie są programistami
> niektóre delfiny są inteligentne

wynika zdanie:

> niektórzy nie programujący komputerów są inteligentni.

Monotoniczność

Logikę nazywamy monotoniczną, jeżeli wnioski wynikające z bazy wiedzy B_1 wynikają także z bazy $B_1 \cup B_2$, czyli

$$\text{jeżeli } B_1 \models \alpha, \text{ to } (B_1 \cup B_2) \models \alpha$$

Logiki wyższego rzędu

Logiki wyższego rzędu pozwalają dodatkowo na kwantyfikacje po relacjach (predykatach) jak i funkcjach.

Logiki wyższego rzędu

- Np. w logice wyższego rzędu można napisać wyrażenie

$$(\forall XY)(X = Y) \equiv ((\forall p)p(X) \equiv p(Y))$$

oznaczające,

Logiki wyższego rzędu

- Np. w logice wyższego rzędu można napisać wyrażenie

$$(\forall XY)(X = Y) \equiv ((\forall p)p(X) \equiv p(Y))$$

  oznaczające,
- że wszystkie obiekty są równe wtedy i tylko wtedy, gdy wszystkie ich własności są równe.

Logiki wyższego rzędu

- lub wyrażenie

$$(\forall f, g)(f = g) \equiv ((\forall X) f(X) = g(X))$$

oznaczające,

Logiki wyższego rzędu

- lub wyrażenie

$$(\forall f, g)(f = g) \equiv ((\forall X) f(X) = g(X))$$

  oznaczające,
- że dwie funkcje są równe wtedy i tylko wtedy, gdy mają takie same wartości dla wszystkich wartości argumentów.

Zagadnienie pełności

Pełność jest własnością systemu dowodzenia mówiącą, że każda formuła prawdziwa daje się udowodnić w tym systemie

Dowód wykorzystujący regułę modus ponens jest pełny w przypadku wyrażeń zapisanych w postaci klauzul Horna.

Zagadnienie pełności

Przykład

Stosując wyłącznie regułę modus ponens wykazać $s(a)$ na podstawie następujących wyrażeń:

$$(\forall X)p(X) \rightarrow q(X)$$
$$(\forall X)\neg p(X) \rightarrow r(X)$$
$$(\forall X)q(X) \rightarrow s(X)$$
$$(\forall X)r(X) \rightarrow s(X)$$

Zagadnienie pełności

Przykład

$$(\forall X)p(X) \rightarrow q(X)$$
$$(\forall X)\neg p(X) \rightarrow r(X)$$
$$(\forall X)q(X) \rightarrow s(X)$$
$$(\forall X)r(X) \rightarrow s(X)$$

Jak łatwo się przekonać, stosując wyłącznie regułę modus ponens dowodu przeprowadzić się nie daje (drugie wyrażenie nie jest klauzulą Horna). Czyli, jeżeli wyrażenia nie są w postaci klauzul Horna, to procedura dowodu oparta wyłącznie o regułę wnioskowania modus ponens nie jest pełna, .

Zagadnienie pełności

Jeżeli pełna procedura dowodzenia twierdzeń matematycznych by istniała i była znana to: (a) wszystkie hipotezy mogłyby być wykazane mechanicznie, (b) cała matematyka mogłaby być wygenerowana z zespołu podstawowych aksjomatów. Pytania o pełność doprowadziły do powstania największych prac matematycznych XX wieku.

Zagadnienie pełności

- Kurt Gödel w 1930 roku podał twierdzenie o pełności w logice pierwszego rzędu: każde wyrażenie wynikające ze zbioru wpz BW może być z niego wyprowadzone przez pewną procedurę R, czyli

$$\text{jeżeli } BW \models \alpha \text{ to } BW \vdash_R \alpha$$

Zagadnienie pełności

- Kurt Gödel w 1930 roku podał twierdzenie o pełności w logice pierwszego rzędu: każde wyrażenie wynikające ze zbioru wpz BW może być z niego wyprowadzone przez pewną procedurę R, czyli

$$\text{jeżeli } BW \models \alpha \text{ to } BW \vdash_R \alpha$$

- Gödel nie podał algorytmu R, stwierdził tylko, że istnieje.

Zagadnienie pełności

- Kurt Gödel w 1930 roku podał twierdzenie o pełności w logice pierwszego rzędu: każde wyrażenie wynikające ze zbioru wpz BW może być z niego wyprowadzone przez pewną procedurę R, czyli

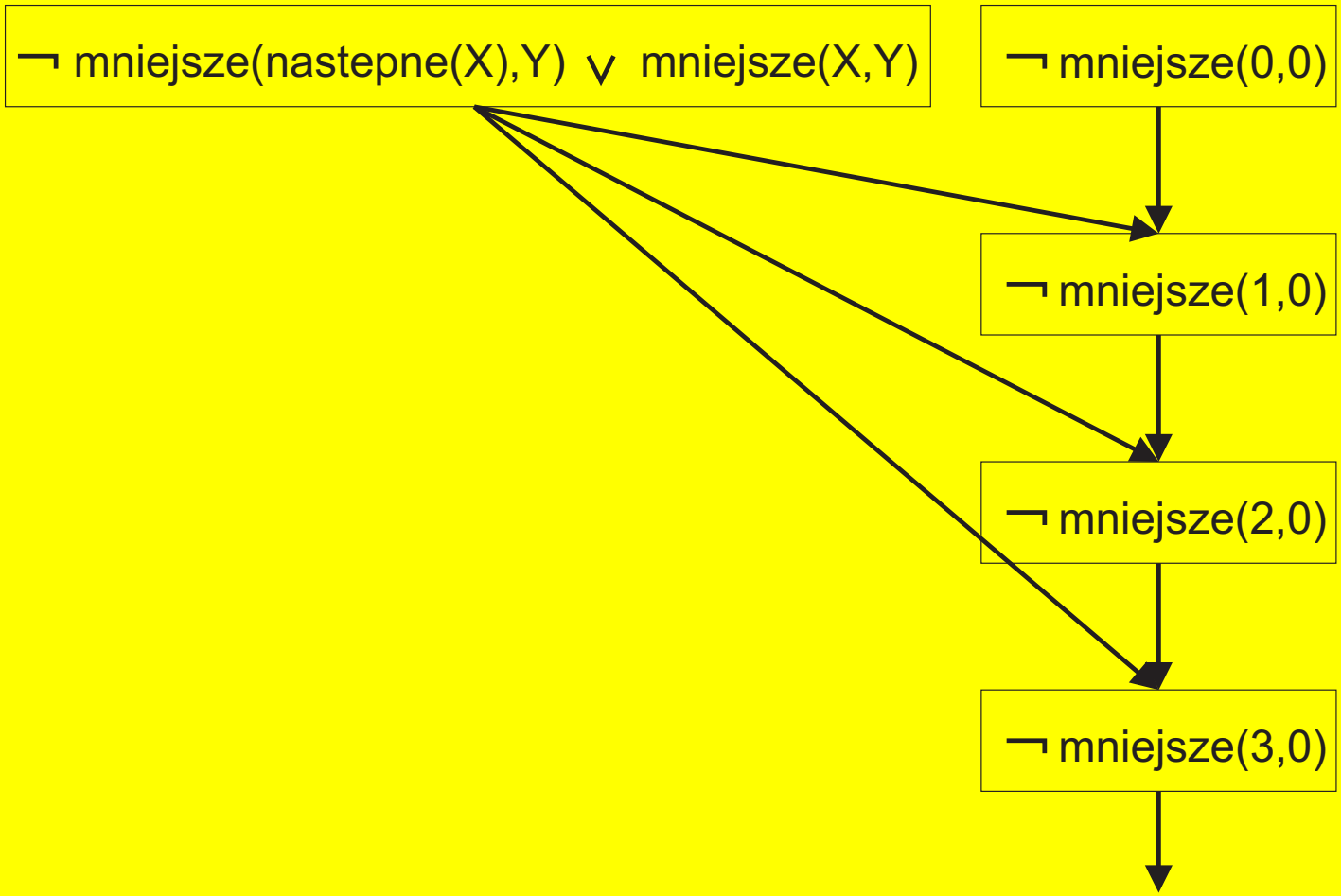
$$\text{jeżeli } BW \models \alpha \text{ to } BW \vdash_R \alpha$$

- Gödel nie podał algorytmu R, stwierdził tylko, że istnieje.
- J.A. Robinson (1965) podał algorytm dowodu oparty o rezolucję i wykazał jego pełność.

Zagadnienie rozstrzygalności

Logika pierwszego rzędu jest *częściowo rozstrzygalna*, tzn. że jeżeli jakieś wyrażenie wynika z danej bazy wiedzy, to w skończonym czasie można to wykazać, natomiast jeżeli nie wynika, to procedura dowodzenia może nigdy się nie zakończyć.
Jest to konsekwencją występowania kwantyf katorów i zagnieżdżonych funkcji co prowadzi do baz o nieskończonych rozmiarach.
Z częściowej rozstrzygalności logiki pierwszego rzędu wynika, że wykazanie sprzeczności bazy wiedzy jest też częściowo rozstrzygalne.

Zagadnienie rozstrzygalności



Przykład niekończacego się wnioskowania

Zagadnienie pełności

Poprzez niewielkie rozszerzenie języka pierwszego rzędu o schemat indukcji Gödel sformułował i udowodnił słynne twierdzenie o niepełności: istnieją zdania arytmetyczne prawdziwe, których nie da się wyprowadzić z przyjętych dla arytmetyki aksjomatów. Twierdzenie Gödla wraz z odkryciem f zyki kwantowej, zasadą nieoznaczoności Heisenberga, problemem stopu w maszynie Turinga oraz odkryciem procesów chaotycznych spowodowało odejście od deterministycznego i przewidywalnego świata Newtona dominującego w f zyce XVIII i XIX wieku.

Język PROLOG a wnioskowanie w LPR

Przykład

Stosując regułę rezolucji należy pokazać, że wpz $(\exists Z)A(Z)$ wynika z

(2) $A(X) \vee \neg P(X) \vee \neg Q(X, Y)$

(3) $A(V) \vee \neg R(W) \vee \neg Q(W, V)$

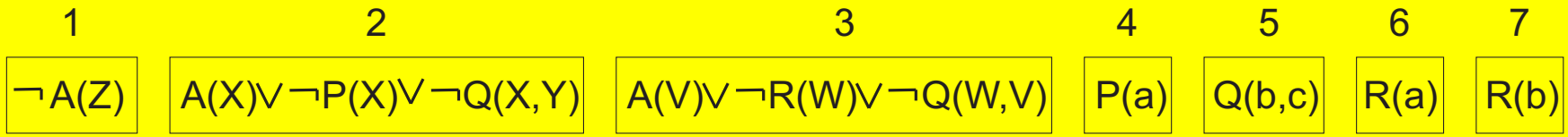
(4) $P(a)$

(5) $Q(b, c)$

(6) $R(a)$

(7) $R(b)$

Język PROLOG a wnioskowanie w LPR



Przykład wnioskowania

Język PROLOG a wnioskowanie w LPR

(2) $A(X) :- P(X), Q(X, Y)$

(3) $A(V) :- R(W), Q(W, V)$

(4) $P(a)$

(5) $Q(b, c)$

(6) $R(a)$

(7) $R(b)$

goal $A(X)$

Język PROLOG a wnioskowanie w LPR

Biorąc pod uwagę, że:

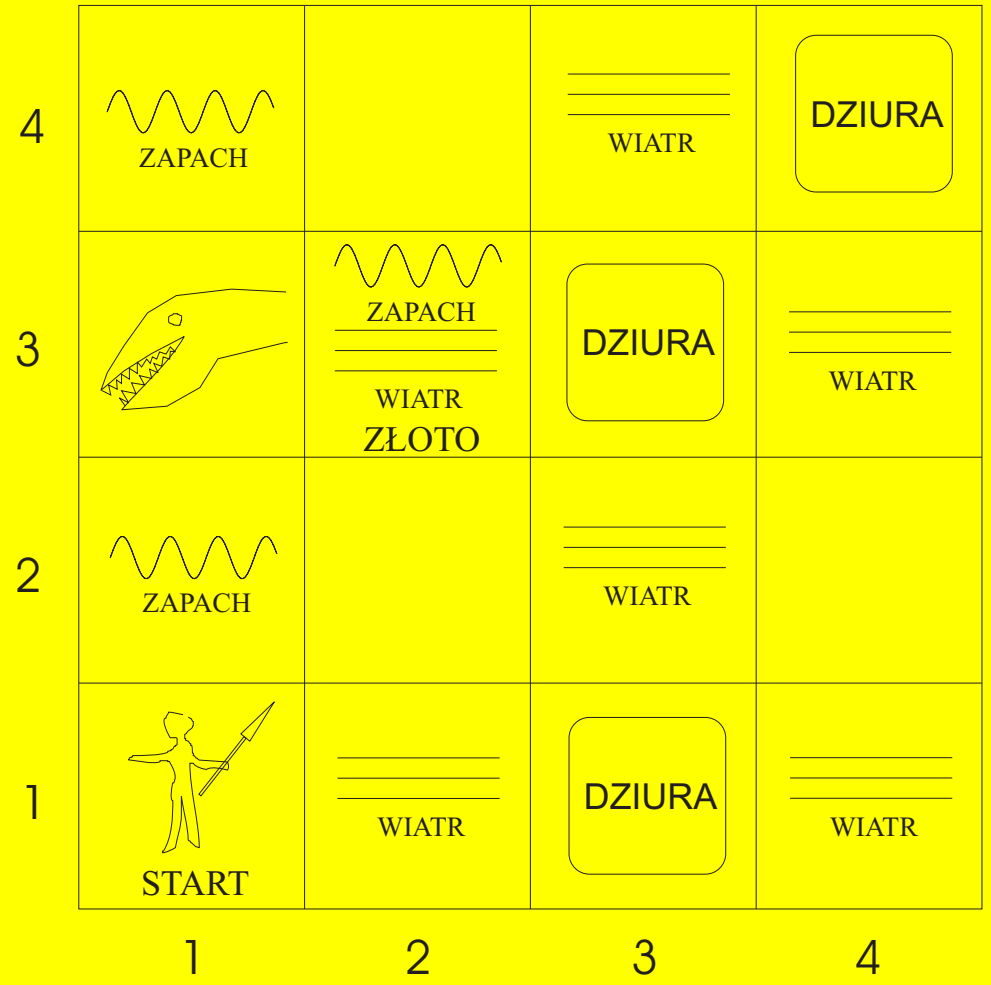
- Klauzule (2)–(7) są klauzulami Horna.
- Przyjęto ustaloną strategię kolejności prób wyznaczania rezolwent polegającą na poszukiwaniu atomów unifikujących się z pierwszym literałem (w kolejności występowania w klauzuli) ostatnio otrzymanej klauzuli, a w przypadku niepowodzenia cofanie się do poprzedniej klauzuli i próba ponownego znalezienia unifikacji jej pierwszego literału.

widać, że klauzule (2)–(7) przedstawione w przykładzie są równoważne klauzulom programu w PROLOG'u oraz strategia wnioskowania jest identyczna z realizowaną w PROLOG'u.

Język PROLOG a wnioskowanie w LPR

Wynika stąd, że wykonywanie programu w PROLOG'u jest równoważne wnioskowaniu w logice pierwszego rzędu przy zastosowaniu reguły rezolucji i strategii wnioskowania w tył.

Świat smoka (wumpus world)



Przykład świata smoka

Świat smoka (wumpus world)

Podczas działania systemu agent odbiera następujące sygnały, podejmuje następujące akcje i ma następujące cele:

- W kwadracie zawierającym smoka i w kwadratach przylegających bokami agent czuje *zapach*.

Świat smoka (wumpus world)

Podczas działania systemu agent odbiera następujące sygnały, podejmuje następujące akcje i ma następujące cele:

- W kwadracie zawierającym smoka i w kwadratach przylegających bokami agent czuje *zapach*.
- W kwadratach przylegających bokami do dziury agent czuje powiew wiatru.

Świat smoka (wumpus world)

Podczas działania systemu agent odbiera następujące sygnały, podejmuje następujące akcje i ma następujące cele:

- W kwadracie zawierającym smoka i w kwadratach przylegających bokami agent czuje *zapach*.
- W kwadratach przylegających bokami do dziury agent czuje powiew wiatru.
- W kwadracie zawierającym złoto agent zauważa blask.

Świat smoka (wumpus world)

Podczas działania systemu agent odbiera następujące sygnały, podejmuje następujące akcje i ma następujące cele:

- W kwadracie zawierającym smoka i w kwadratach przylegających bokami agent czuje *zapach*.
- W kwadratach przylegających bokami do dziury agent czuje powiew wiatru.
- W kwadracie zawierającym złoto agent zauważa blask.
- Gdy agent usiłuje przejść przez ściane ograniczającą system, to czuje uderzenie.

Świat smoka (wumpus world)

Podczas działania systemu agent odbiera następujące sygnały, podejmuje następujące akcje i ma następujące cele:

- W kwadracie zawierającym smoka i w kwadratach przylegających bokami agent czuje *zapach*.
- W kwadratach przylegających bokami do dziury agent czuje powiew wiatru.
- W kwadracie zawierającym złoto agent zauważa blask.
- Gdy agent usiłuje przejść przez ściane ograniczającą system, to czuje uderzenie.
- Gdy smok zostaje zabity, to wydaje przenikliwy dźwięk, który jest słyszalny w każdym kwadracie

Świat smoka (wumpus world)

- Agent odbiera docierające do niego sygnały w postaci list pięciu symboli. Na przykład, jeżeli jest zapach i wiatr ale nie ma blasku, uderzenia i przenikliwego dźwięku, agent otrzyma listę: [*Zapach*, *Wiatr*, *Nic*, *Nic*, *Nic*]. Agent nie postrzega swojego położenia.

Świat smoka (wumpus world)

- Agent odbiera docierające do niego sygnały w postaci list pięciu symboli. Na przykład, jeżeli jest zapach i wiatr ale nie ma blasku, uderzenia i przenikliwego dźwięku, agent otrzyma listę: [*Zapach*, *Wiatr*, *Nic*, *Nic*, *Nic*]. Agent nie postrzega swojego położenia.
- Agent może podjąć następujące akcje: przesunięcie się o jeden kwadrat do przodu, obrót w prawo o 90^0, obrót w lewo o 90^0, chwycenie obiektu znajdującego się w tym samym kwadracie, w którym znajduje się agent. Ponadto agent może wystrzelić strzałę, która leci wzdłuż lini prostej i albo traf i zabije smoka, albo uderzy w ścianę. Agent ma tylko jedną strzałe, więc tylko jedna akcja strzału może mieć jakiś skutek.

Świat smoka (wumpus world)

- Agent może podjąć akcje wyjścia z jaskini; co może być zrealizowane, gdy agent znajduje się w kwadracie $[1, 1]$.

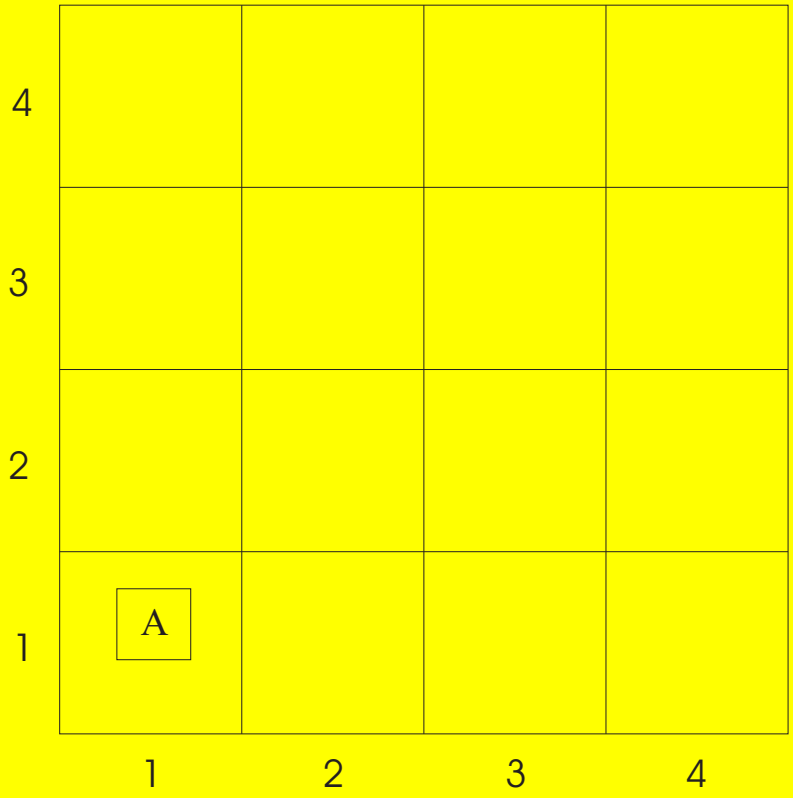
Świat smoka (wumpus world)

- Agent może podjąć akcje wyjścia z jaskini; co może być zrealizowane, gdy agent znajduje się w kwadracie $[1, 1]$.
- Agent ginie jeżeli wejdzie do kratki zawierającej dziurę lub żywego smoka.

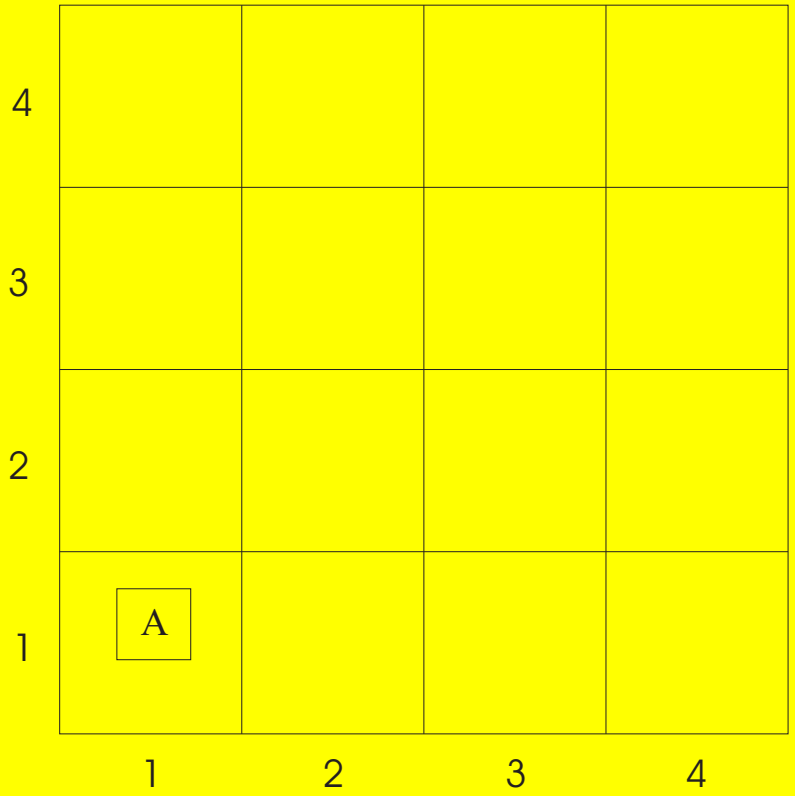
Świat smoka (wumpus world)

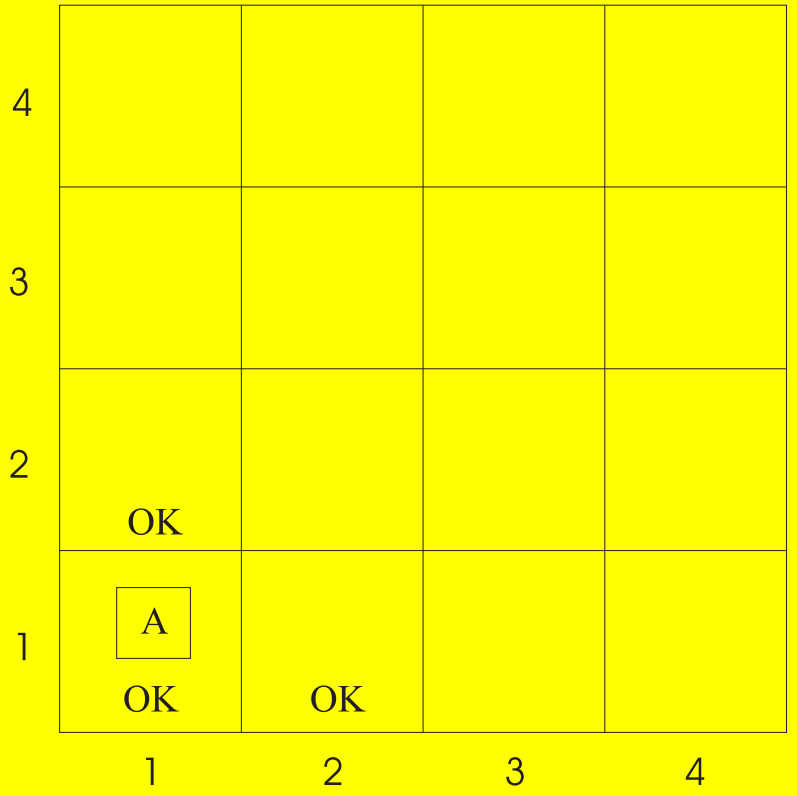
- Agent może podjąć akcje wyjścia z jaskini; co może być zrealizowane, gdy agent znajduje się w kwadracie $[1, 1]$.
- Agent ginie jeżeli wejdzie do kratki zawierającej dziurę lub żywego smoka.
- Celem agenta jest znalezienie złota i wyniesienie go z jaskini tak szybko jak to jest możliwe. W jednorazowej próbie, za wyniesienie złota z jaskini przyznane zostaje 1000 punktów, jedna podjęta akcja kosztuje 1 punkt zaś utrata życia przez agenta kosztuje 10000 punktów.

Wnioskowanie w świecie smoka

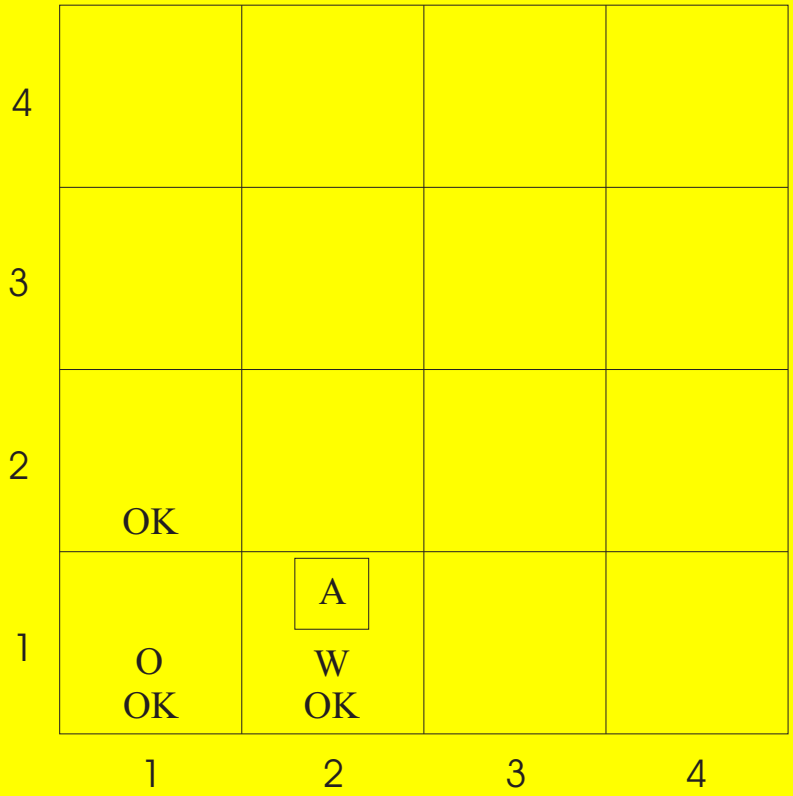


Wnioskowanie w świecie smoka

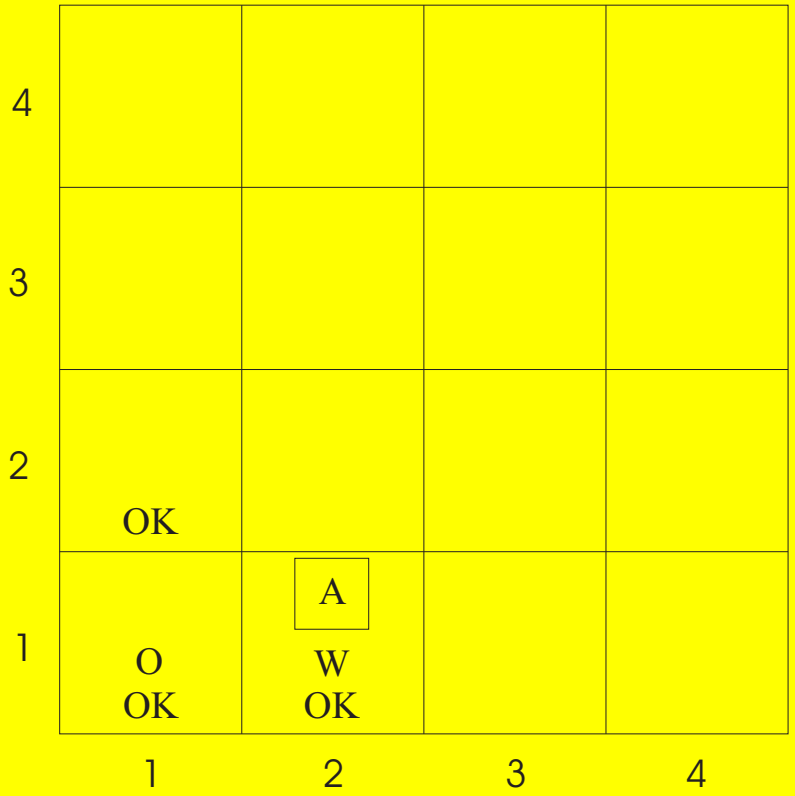


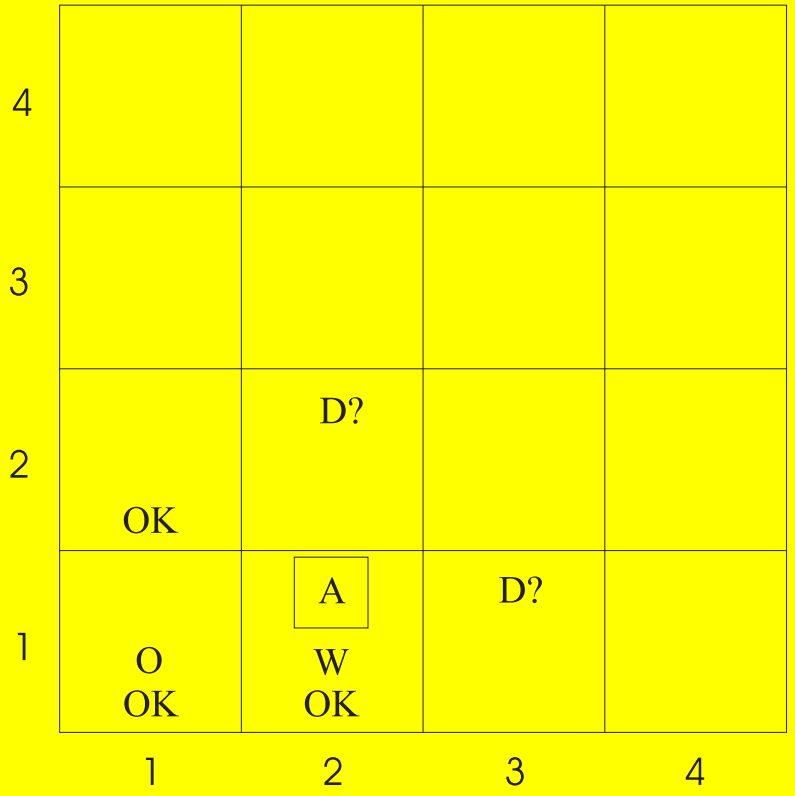


Wnioskowanie w świecie smoka

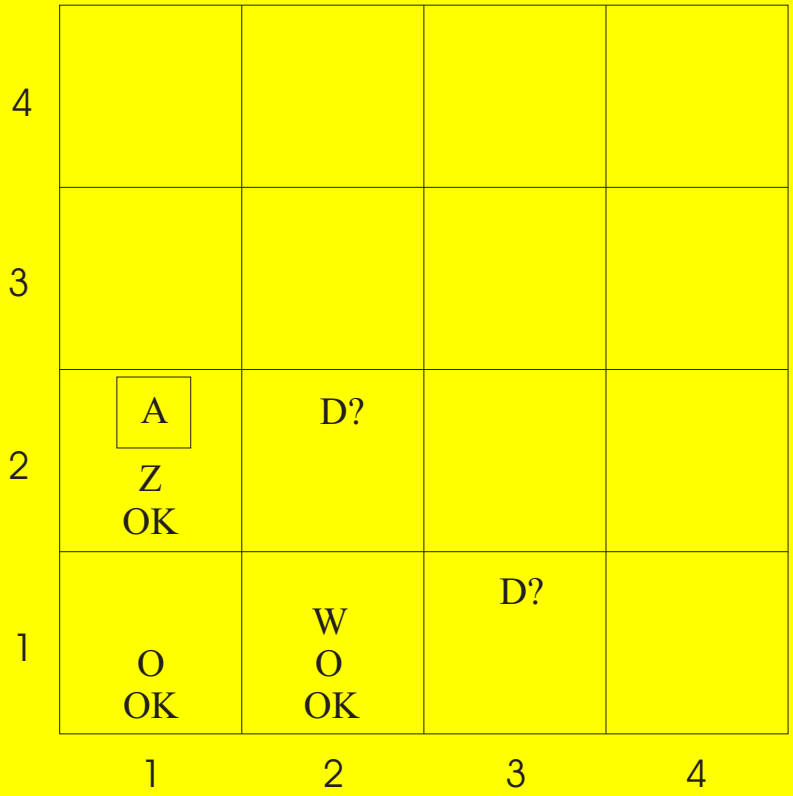


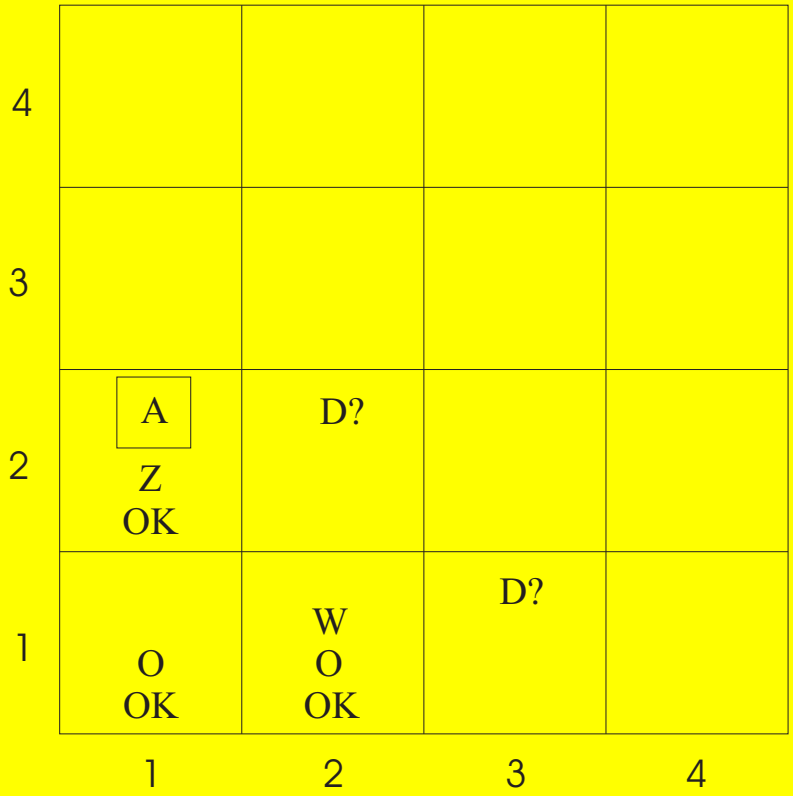
Wnioskowanie w świecie smoka

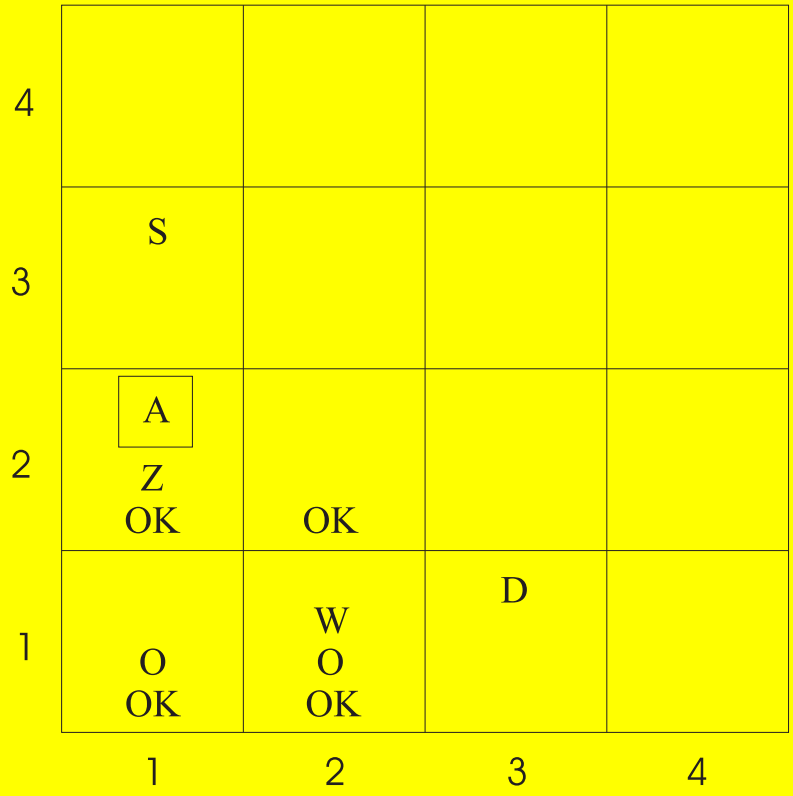




Wnioskowanie w świecie smoka




4
3
2
1
A
Z
OK
D?
O
OK
W
O
OK
D?
1
2
3
4


4
3
2
1
S
A
Z
OK
OK
O
OK
W
O
OK
D
1
2
3
4

Wnioskowanie w świecie smoka

„Zapach występuje w kwadracie wtedy i tylko wtedy, gdy przynajmniej w jednym sąsiednim kwadracie znajduje się smok“ Na podstawie tego zdania mamy

$$(\forall X, Y)[ws(X, Y) \wedge z(X, Y)] \equiv$$
$$\equiv (\exists X_s, Y_s)[ws(X_s, Y_s) \wedge ks(X, Y, X_s, Y_s) \wedge s(X_s, Y_s)]$$

gdzie predykaty ws, z, ks i s określają odpowiednio znajdowanie sie w systemie, zapach, kwadrat sąsiedni i obecność smoka.

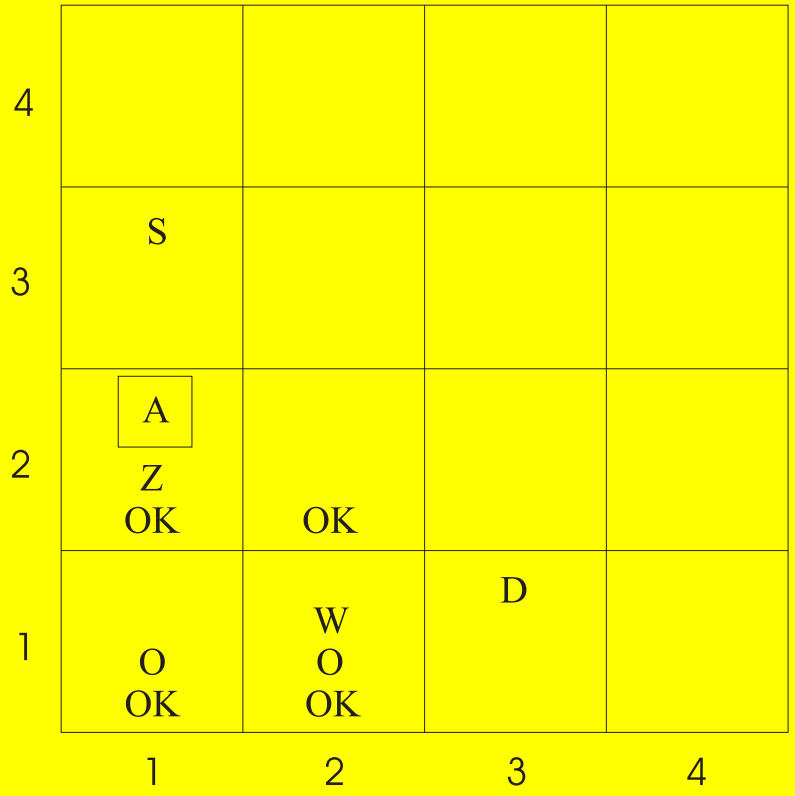
Wnioskowanie w świecie smoka

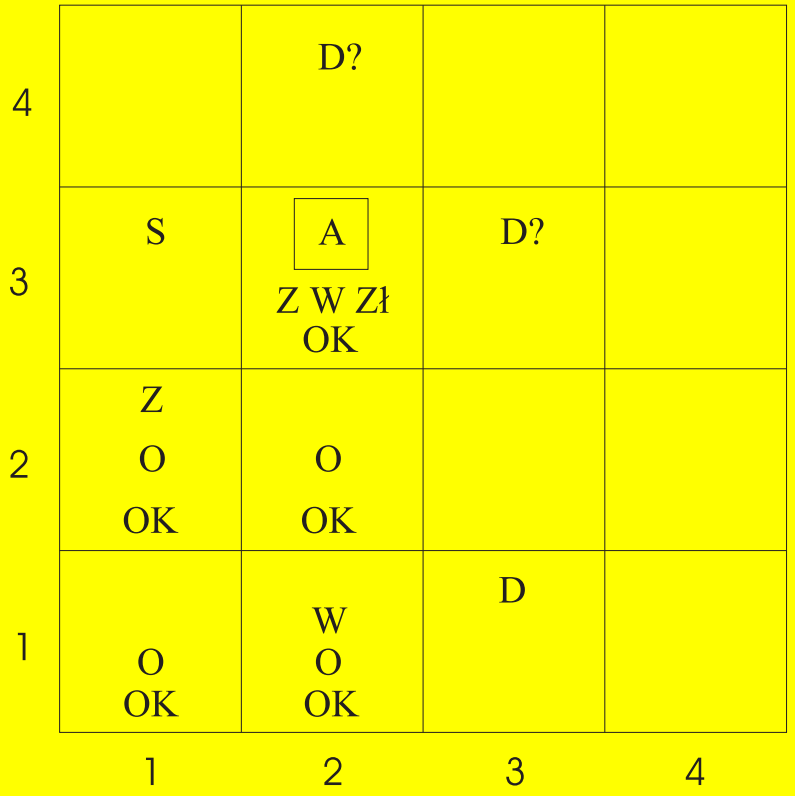
Nie sprawdzając znajdowania sie współrzędnych wewnątrz systemu, oraz zapisując sąsiedztwo w jawny sposób i nie piszac jawnie kwantyf katorów można na podstawie powyższego wyrażenia napisać dwie równoważne reguły:

$$\neg z(X,Y) \rightarrow \neg s(X,Y) \wedge \neg s(X-1,Y) \wedge \neg s(X+1,Y) \wedge \neg s($$

$$z(X,Y) \rightarrow s(X,Y) \vee s(X-1,Y) \vee s(X+1,Y) \vee s(X,Y-1$$

Wnioskowanie w świecie smoka





Dalsze kroki agenta w świecie smoka: (a) stan po trzecim kroku i percepcji [*Zapach, Nic, Nic, Nic, Nic*], (b) po piątym kroku i percepcji [*Zapach, Wiatr, Blask, Nic, Nic*]