

Wprowadzenie do logiki

KPI KUL

ai.kul.pl

18 września 2022



Spis treści

1 Klasyczny rachunek zdań

- Rzut oka wstecz
- Metoda tablic analitycznych
- Metalogiczne własności KRZ

2 Węższy rachunek predykatów

- Motywacja
- Język WRP
- Metoda tablic analitycznych
- Semantyka dla WRP
- Metalogiczne własności WRP
- Systemy automatycznego dowodzenia twierdzeń

3 Logiki opisowe

- Logika opisowa \mathcal{ALC}
- Reprezentacja relacji w logice
- Logika opisowa \mathcal{SROIQ}

- dlaczego?
- co?
- jak?
- dla-czego-nie?

Skróty

- KRZ
- PZW_{KRZ}
- WRP
- PZW_{WRP}

PZW_{KRZ} w notacji BNF

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \phi \equiv \phi \quad (1)$$

PZW_{KRZ} w notacji BNF (2)

$< zero_digit > ::= "0"$ (2)

$< nonzero_digit > ::= "1" "2" "4" "5" "6" "7" "8" "9"$ (3)

$< digit > ::= < zero_digit > | < nonzero_digit >$ (4)

$< index > ::= < digit > | < nonzero_digit > < index >$ (5)

$< atom > ::= "pindex >$ (6)

$< \phi > ::= < atom > |$

$"(\cdot \cdot \cdot \neg < \phi > \cdot \cdot \cdot)$

$"(< \phi > \wedge < \phi >)$

$"(< \phi > \vee < \phi >)$

$"(< \phi > \rightarrow < \phi >)$

$"(< \phi > \equiv < \phi >)"$ (7)

Co każdy student chciałby wiedzieć o metodzie tablic analitycznych, ale wstydzi się zapytać

- Praw KRZ szukamy poprzez konstrukcję tablic analitycznych.
- Metoda tablic jest zmodyfikowaną wersją skróconej metody zero-jedynkowej.
- Tablica analityczna jest binarnym drzewem znakowanym wyrażeniami PZW_{KRZ} .

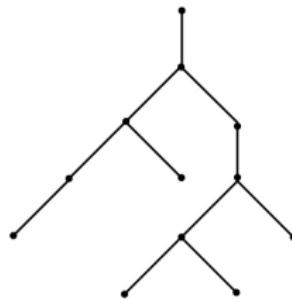
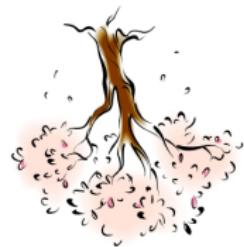
Metody KRZ - porównanie

Metoda założeniowa
trzeba myśleć
nadaje się do WRP

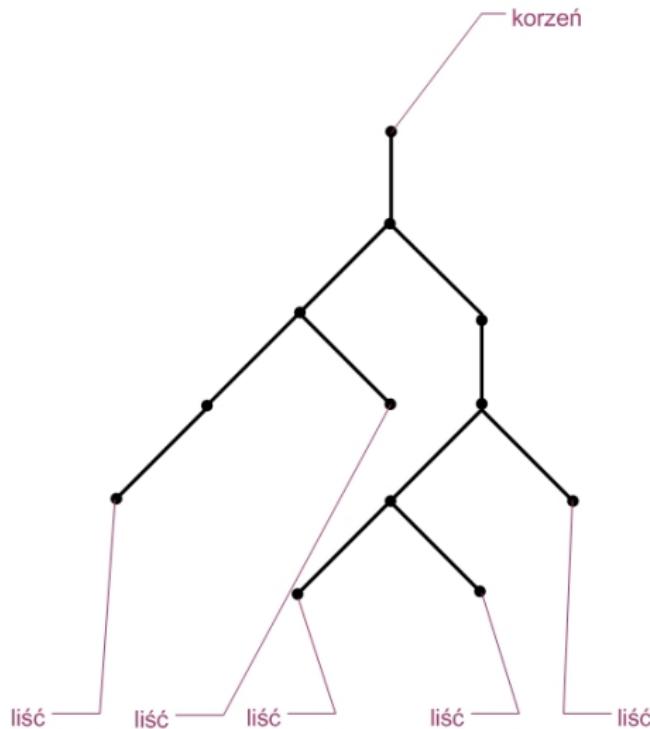
Metoda 01-owa
nie trzeba myśleć
nie nadaje się do WRP

Metoda tablic
nie trzeba myśleć
nadaje się do WRP

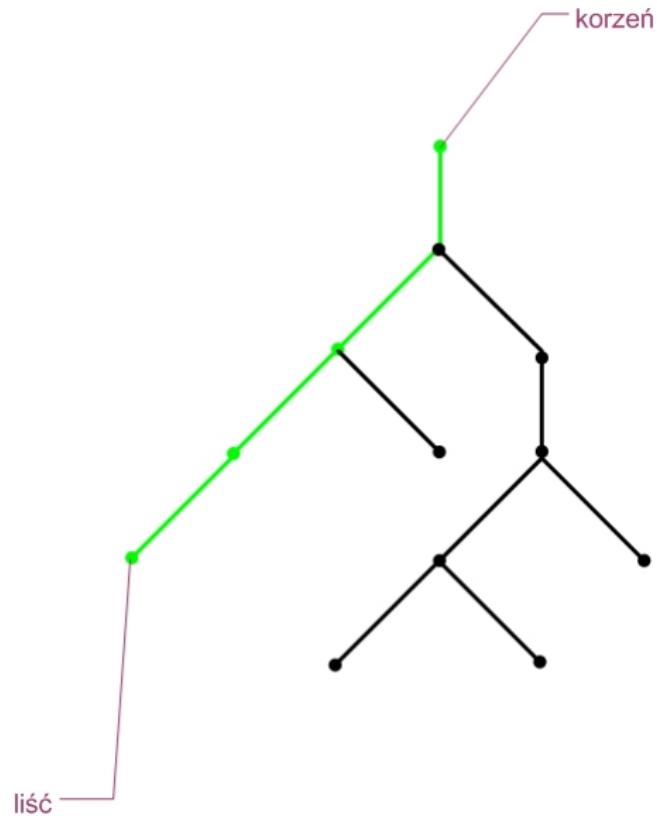
Drzewa i drzewa



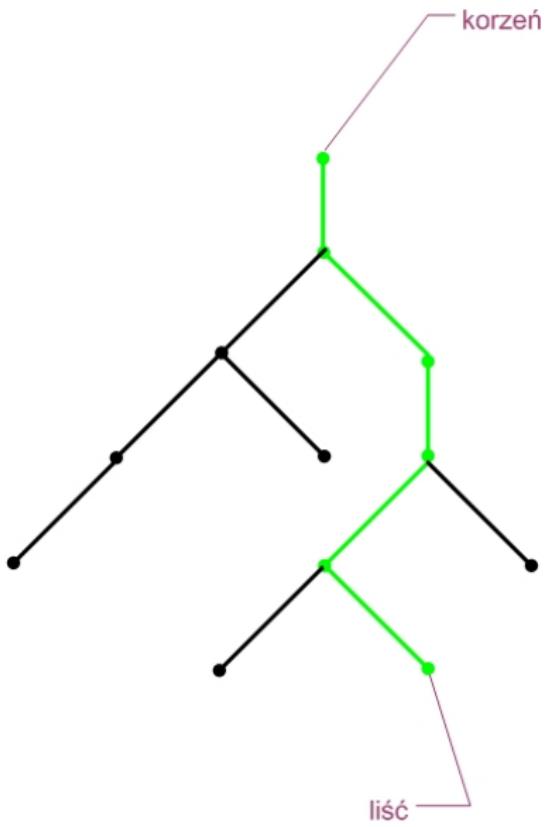
Drzewa - korzenie i liście



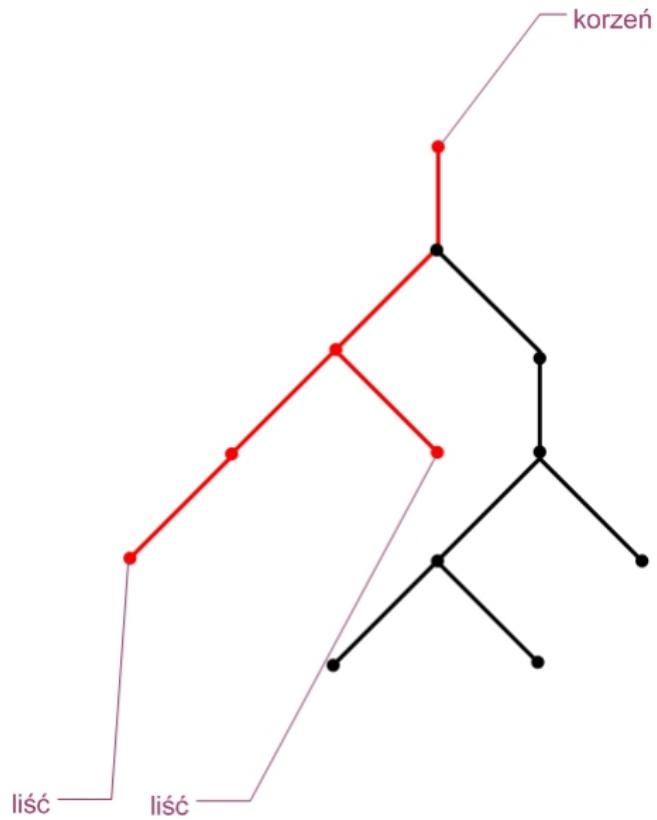
Drzewa - gałęzie, korzenie i liście



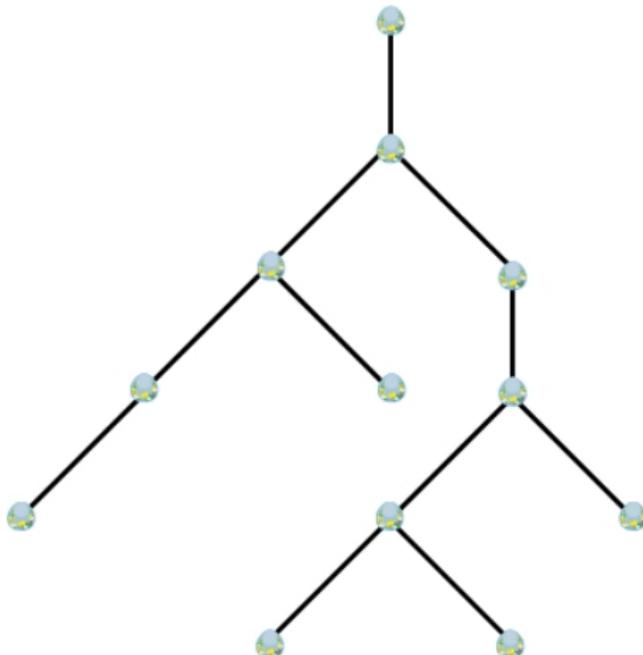
Drzewa - gałęzie, korzenie i liście (2)



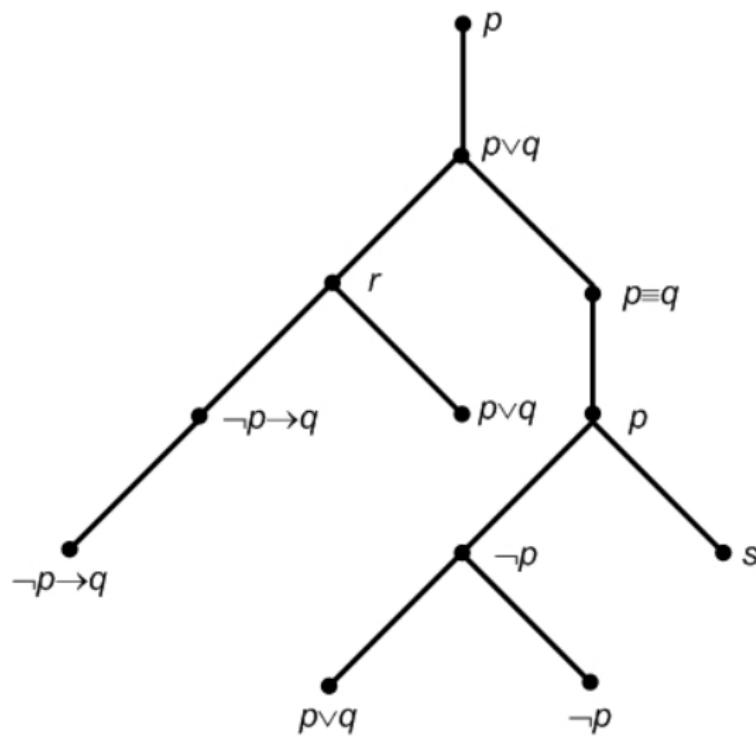
Drzewa - gałęzie, korzenie i liście (3)



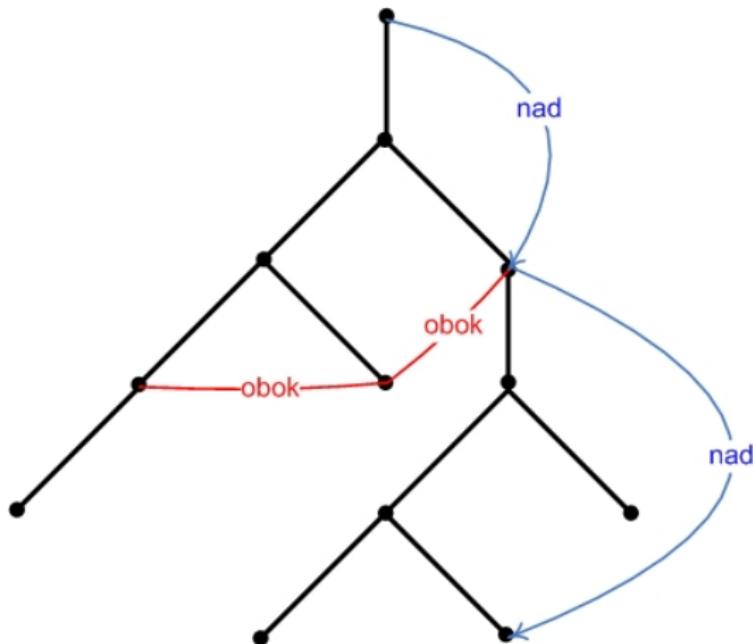
Drzewa znakowane



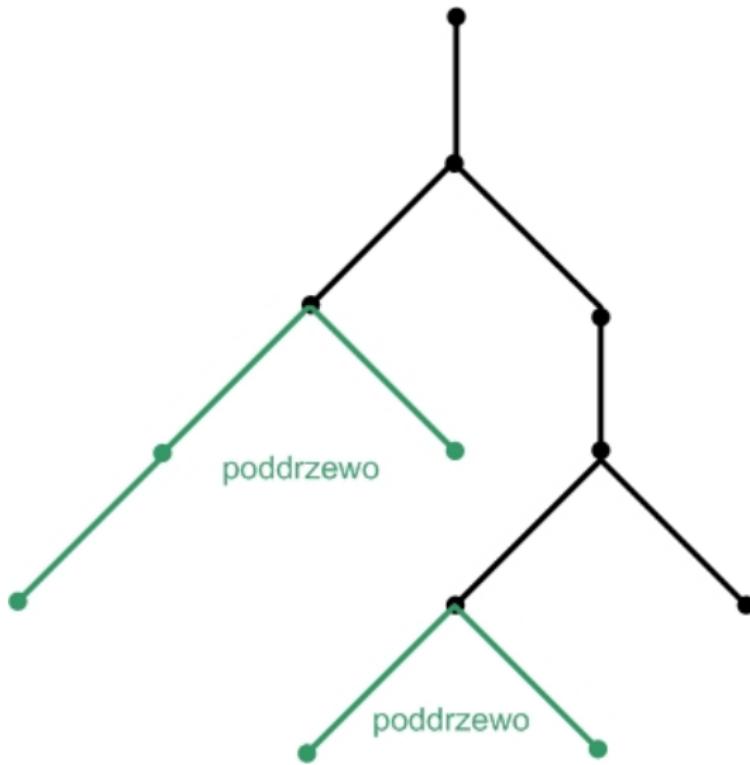
Drzewa znakowane (2)



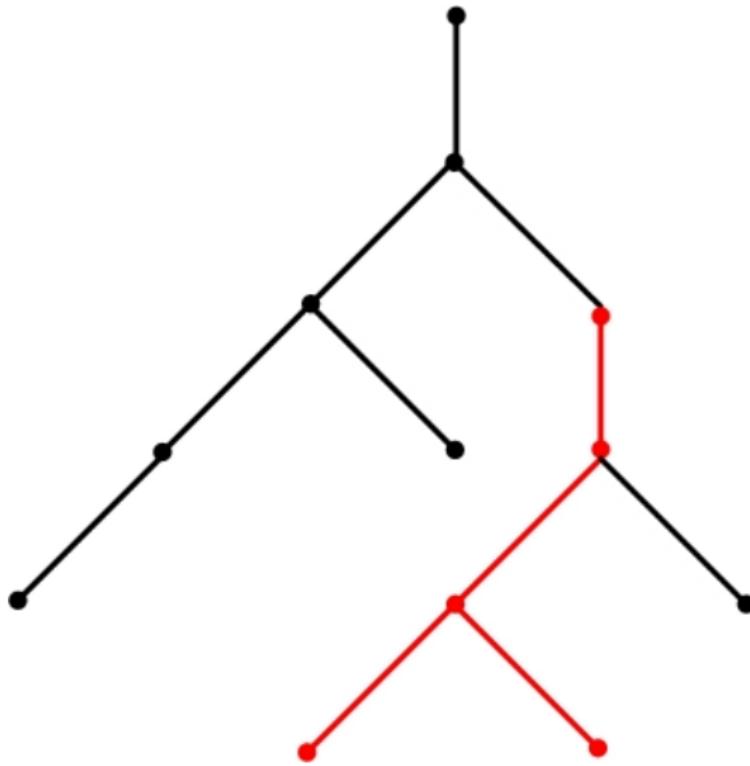
Drzewa - nad/pod i obok



Drzewa i poddrzewa



Drzewa i poddrzewa (2)



Tablice analityczne – skąd je brać?

- Tablice analityczne składają się z atomowych tablic analitycznych.
- Tworzenie tablic analitycznych polega na dodawaniu tablic atomowych do istniejącej tablicy.
- W KRZ mamy jedenaście tablic atomowych:
 - ▶ T: p
 - ▶ TN: $\neg p$
 - ▶ TON:
$$\begin{array}{c} \neg\neg\phi \\ | \\ \phi \end{array}$$

Tablice analityczne – skąd je brać? (2)

- TOK:

$$\begin{array}{c} \phi \wedge \psi \\ | \\ \phi \\ | \\ \psi \end{array}$$

- TOA:

$$\begin{array}{c} \phi \vee \psi \\ \swarrow \quad \searrow \end{array}$$

- TOC:

$$\begin{array}{c} \phi \rightarrow \psi \\ \neg\phi \quad \psi \end{array}$$

Tablice analityczne – skąd je brać? (3)

- TOE:

$$\phi \equiv \psi$$

$$\begin{array}{ccc} & \diagup & \diagdown \\ \phi & & \neg\phi \\ | & & | \\ \psi & & \neg\psi \end{array}$$

- TONK:

$$\neg(\phi \wedge \psi)$$

$$\begin{array}{ccc} & \diagup & \diagdown \\ \neg\phi & & \neg\psi \end{array}$$

- TONA:

$$\neg(\phi \vee \psi)$$

$$\begin{array}{c} | \\ \neg\phi \\ | \\ \neg\psi \end{array}$$

Tablice analityczne – skąd je brać? (4)

- TONC:

$$\begin{array}{c} \neg(\phi \rightarrow \psi) \\ | \\ \phi \\ | \\ \neg\psi \end{array}$$

- TONE:

$$\begin{array}{ccc} & \swarrow & \searrow \\ \phi & & \neg\phi \\ | & & | \\ \neg\psi & & \psi \end{array}$$

Zredukowane tablice analityczne

- Tablice analityczne konstrujemy tak długo, aż uzyskamy zredukowaną ich postać.
- Jeżeli ϕ jest wyrażeniem KRZ, tj. gdy należy do PZW_{KRZ} , to symbol T_ϕ oznacza drzewo atomowe, które ma w korzeniu ϕ .

Definicja

Niech T będzie tablicą analityczną.

- ① wyrażenie ϕ jest zredukowane w T jeżeli drzewo T'_ϕ jest poddrzewem T .
- ② Tablica analityczna T jest zredukowana jeżeli wszystkie wyrażenia z T są zredukowane w T .

Zamknięte i otwarte tablice analityczne

Po uzyskaniu tablicy zredukowanej sprawdzamy, czy uzyskana tabela jest otwarta, czy zamknięta.

Definicja

Tablica analityczna jest zamknięta, jeżeli każda jej gałąź zawiera dwa wyrażenia sprzeczne: $\phi, \neg\phi$.

Tablica analityczna jest otwarta, jeżeli nie jest zamknięta.

Fact

Wyrażenie $\phi \in \mathbb{PZW}_{\text{KRZ}}$ jest prawem logiki wtedy i tylko wtedy, gdy istnieje zamknięta tablica analityczna dla $\neg\phi$.

Metoda tablic analitycznych - krok po kroku

Czy $\phi \in \mathbb{PZW}_{\text{KRZ}}$ jest prawem logiki?

- ① Wstaw $\neg\phi$ jako korzeń tablicy analitycznej T !
- ② Korzystając z atomowych tablic analitycznych zredukuj T !
- ③ Czy T jest zamknięta?
 - ▶ TAK $\implies \phi$ jest prawem logiki.
 - ▶ NIE $\implies \phi$ nie jest prawem logiki.

Metoda tablic analitycznych - przykłady

Czy wyrażenie $(p \rightarrow q) \rightarrow p \vee q$ jest prawem logiki?

Metoda tablic analitycznych - przykłady

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

Metoda tablic analitycznych - przykłady

$$\begin{array}{c} \neg((p \rightarrow q) \rightarrow p \vee q) \\ | \\ p \rightarrow q \\ | \\ \neg(p \vee q) \end{array}$$

Metoda tablicy analitycznych - przykłady

$$\begin{array}{c} \neg((p \rightarrow q) \rightarrow p \vee q) \\ | \\ p \rightarrow q \\ | \\ \neg(p \vee q) \\ | \\ \neg p \\ | \\ \neg q \end{array}$$

Metoda tablicy analitycznych - przykłady

$$\begin{array}{c} \neg((p \rightarrow q) \rightarrow p \vee q) \\ | \\ p \rightarrow q \\ | \\ \neg(p \vee q) \\ | \\ \neg p \\ | \\ \neg q \\ \swarrow \searrow \\ \neg p \quad q \end{array}$$

Metoda tablicy analitycznych - przykłady (2)

Ponieważ następująca gałąź w tej tablicy *nie* zawiera wyrażeń sprzecznych:

$$\begin{array}{c} \neg((p \rightarrow q) \rightarrow p \vee q) \\ | \\ p \rightarrow q \\ | \\ \neg(p \vee q) \\ | \\ \neg p \\ | \\ \neg q \\ \neg p \wedge \end{array}$$

zatem wyrażenie $(p \rightarrow q) \rightarrow p \vee q$ *nie* jest prawem logiki, bo uzyskana tablica analityczna jest otwarta.

Metoda tablic analitycznych - przykłady (3)

Czy wyrażenie $q \rightarrow (p \rightarrow q)$ jest prawem logiki?

Metoda tablic analitycznych - przykłady (3)

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

Metoda tablic analitycznych - przykłady (3)

$$\begin{array}{c} \neg(q \rightarrow (p \rightarrow q)) \\ | \\ q \\ | \\ \neg(p \rightarrow q) \end{array}$$

Metoda tablicy analitycznych - przykłady (3)

$$\begin{array}{c} \neg(q \rightarrow (p \rightarrow q)) \\ | \\ q \\ | \\ \neg(p \rightarrow q) \\ | \\ p \\ | \\ \neg q \end{array}$$

Metoda tablic analitycznych - przykłady (3)

Ponieważ jedyna gałąź w tej tablicy zawiera wyrażenia *sprzeczne*: q , $\neg q$, zatem wyrażenie $q \rightarrow (p \rightarrow q)$ jest prawem logiki.

Modele i kontrmodele

- model i kontrmodel formuły
- modele a drzewa

Automatyzacja metody drzew/tablic

- <https://www.umsu.de/trees/>

Metalogiczne własności KRZ

- ① KRZ jest niesprzeczny
- ② KRZ jest (zu)pełny
- ③ KRZ jest rozstrzygalny

Tezy i tautologie w logice

- *Teza* jest to wyrażenie zdaniowe, dla którego istnieje dowód ($\vdash \phi$).
- *Tautologia* jest to (zawsze) prawdziwe wyrażenie zdaniowe ($\models \phi$).

Niesprzecznośc teorii logicznej

- Teoria logiczna jest to zbiór tez ("zamknięty" ze względu na dowody.)
- Teoria logiczna jest *niesprzeczna* jeżeli wśród jej tez nie ma wyrażeń sprzecznych: ϕ i $\neg\phi$.

Zdrowie i pełność teorii logicznej

- Teoria logiczna jest zdrowa (*sound*), gdy każda jej teza jest tautologią.
- Teoria logiczna jest pełna (*complete*), gdy każda jej tautologia jest tezą.

Rozstrzygalność teorii logicznej

- Teoria logiczna jest *rozstrzygalna*, gdy istnieje metoda, która w skończonej liczbie ścisłe określonych kroków jest w stanie stwierdzić o każdym wyrażeniu, czy jest ono tezą tej teorii czy nie.
- Teoria logiczna jest rozstrzygalna, gdy zbiór jej tez może zostać obliczony przez jakąś maszynę Turinga.

Spis treści

1 Klasyczny rachunek zdań

- Rzut oka wstecz
- Metoda tablic analitycznych
- Metalogiczne własności KRZ

2 Węższy rachunek predykatów

- Motywacja
- Język WRP
- Metoda tablic analitycznych
- Semantyka dla WRP
- Metalogiczne własności WRP
- Systemy automatycznego dowodzenia twierdzeń

3 Logiki opisowe

- Logika opisowa \mathcal{ALC}
- Reprezentacja relacji w logice
- Logika opisowa \mathcal{SROIQ}

Poza KRZ

Pewne wnioskowania dedukcyjne są poza zasięgiem KRZ:

- wnioskowania łatwe, np.:

Żaden gryzoń nie jest gadem.

Nieprawda, że niektóre gady są gryzoniami.

- wnioskowania trudne, np.:

Dłużnicy naszych dłużników są naszymi dłużnikami.

Każdy jest dłużnikiem swego każdego wierzyciela.

Każdy jest wierzycielem swego każdego dłużnika.

Wierzyciele naszych wierzycieli są naszymi wierzycielami.

Porównanie języków

Wiedza	KRZ	sylogistyka	WRP
Jan jest bogaty	p	-	$A(a)$
Warszawa jest stolicą Polski	p	-	$P(a, b)$
Jan jest bogaty lub uczciwy	$p \vee q$	-	$A(a) \vee B(a)$
Anna lub Jan są bogaci	$p \vee q$	-	$A(a) \vee A(b)$
Każdy student jest uczciwy	p	SaP	$\forall x (C(x) \rightarrow B(x))$
Żaden student nie jest uczciwy	p	SeP	$\forall x (C(x) \rightarrow \neg B(x))$
Każdy student jest uczciwy lub odważny	p	-	$\forall x (C(x) \rightarrow B(x) \vee D(x))$
Każdy syn Jana jest wrogiem Piotra	p	-	$\forall x (P(x, a) \rightarrow Q(x, b))$

Tabela: Wiedza w językach logiki

Predykaty

predykaty:

- jednoargumentowe:

- ▶ ... umiera,
- ▶ ... jest nieparzysta.

- wieloargumentowe:

- ▶ ... < ... ,
- ▶ ... ∈ ... ,
- ▶ ... kocha ... ,
- ▶ ... leży pomiędzy ... a

Definicja

Alfabet WRP zawiera:

- ① nieskończony zbiór zmiennych indywidualnych:
 $x, y, z, \dots, x_1, y_1, z_1, \dots, x_2, \dots,$
- ② nieskończony zbiór stałych indywidualnych:
 $a, b, c, \dots, a_1, b_1, c_1, \dots, a_2, \dots,$
- ③ nieskończony zbiór predykatów:
 - ▶ jednoargumentowych: $A, B, C, \dots, A_1, \dots,$
 - ▶ wieloargumentowych: $P, Q, R, \dots, P_1, \dots,$
- ④ zbiór funktorów prawdziwościowych: $\neg, \wedge, \vee, \rightarrow, \equiv,$
- ⑤ zbiór kwantyfikatorów: $\forall, \exists,$
- ⑥ nawiasy: $(,).$

Język WRP

Sumę zbiorów:

- ① zmiennych indywidualnych
- ② stałych indywidualnych

będziemy nazywać zbiorem *wyrażeń nazwowych* WRP.

Dowolny ciąg elementów alfabetu WRP jest *napisem* (WRP).

Język WRP(2)

Definicja

Zbiór poprawnie zbudowanych wyrażeń WRP (skrót: PZW_{WRP}) jest najmniejszym zbiorem napisów spełniającym poniższe warunki:

- ① Jeżeli $\delta^{(n)}$ jest predykatem n -argumentowym ($n \geq 1$), a $\alpha_1, \alpha_2, \dots, \alpha_n$ są wyrażeniami nazwowymi, to $\delta^{(n)}(\alpha_1, \alpha_2, \dots, \alpha_n) \in \text{PZW}_{\text{WRP}}$.
- ② Jeżeli $\phi \in \text{PZW}_{\text{WRP}}$, to $\neg\phi \in \text{PZW}_{\text{WRP}}$.
- ③ Jeżeli $\phi, \psi \in \text{PZW}_{\text{WRP}}$, to
 $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi), (\phi \equiv \psi) \in \text{PZW}_{\text{WRP}}$.
- ④ Jeżeli $\phi \in \text{PZW}_{\text{WRP}}$ i α jest zmienną indywidualną, to $\forall\alpha(\phi)$ i $\exists\alpha(\phi) \in \text{PZW}_{\text{WRP}}$.

Alfabet WRP – notacja BNF

$< zero_digit > ::= "0"$ (8)

$< nonzero_digit > ::= "1" | "2" | "4" | "5" | "6" | "7" | "8" | "9"$ (9)

$< digit > ::= < zero_digit > | < nonzero_digit >$ (10)

$< index > ::= < digit > | < nonzero_digit > < index >$ (11)

$< const > ::= "a" < index >$ (12)

$< var > ::= "x" < index >$ (13)

$< \alpha > ::= < const > | < var >$ (14)

$< seq > ::= < \alpha > | < \alpha > , " < seq >$ (15)

$< pred > ::= "P" < index >$ (16)

Język WRP – notacja BNF

$<atom> ::= <pred> "(" <seq> ")"$ (17)

$<\phi> ::= <atom> |$

" \forall " $<var>$ ϕ |

" \exists " $<var>$ ϕ |

" \neg " $<\phi>$ |

$<\phi> \wedge <\phi>$ |

$<\phi> \vee <\phi>$ |

$<\phi> \rightarrow <\phi>$ |

$<\phi> \equiv <\phi>$

(18)

O zmiennych i stałych - ponownie

- zmienne:

- ▶ reprezentują wyrażenia/stałe
- ▶ występują w językach:
 - ★ matematyki i nauk ścisłych
 - ★ logiki
 - ★ językach programowania
- ▶ nie występują w językach etnicznych
 - ★ coś, ktoś, gdzieś, itp.
 - ★ Coś zjadło mój ser.

- stałe:

- ▶ nic nie reprezentują
- ▶ występują w każdym języku
- ▶ występują w językach etnicznych
 - ★ Bruno zjadł mój ser.

Notacja nawiasowa zredukowana w WRP

W ciągu symboli: kwantyfikator: \forall lub \exists , \neg , \wedge , \vee , \rightarrow , \equiv , każdy symbol wiąże krócej (=silniej) niż symbole występujące po nim.

Język WRP - przykłady

Tabela: Przykłady poprawnie i niepoprawnie zbudowanych PZW_{WRP}

PZW_{WRP}	non- PZW_{WRP}
$A(x)$	$A(x, x)$
$P(a, b)$	$P()$
$\forall x A(x)$	$A(\forall x)$
$\forall x \forall x A(x)$	
$\forall x \exists y P(z_1, z_2)$	$\forall x \exists y P(z_1,)$

Metoda tablic analitycznych w WRP

- rozszerzenie metody tablic analitycznych dla KRZ
- atomowe tablice dla WRP = atomowe tablice dla KRZ +
 - ① TO \forall – powtarzalna,
 - ② TO \exists – niepowtarzalna,
 - ③ TON \forall – niepowtarzalna,
 - ④ TON \exists – powtarzalna.

Ku nowym regułom - krok po kroku

- zasięg (rażenia) kwantyfikatorów
- wolność i niewola zmiennych
- operacja podstawiania

Zasięg kwantyfikatorów

Definicja

W wyrażenach $\forall \alpha \phi$ i $\exists \alpha \phi$ wyrażenie ϕ jest zasięgiem kwantyfikatora: ogólnego lub szczegółowego.

Wolność i niewola zmiennych

Definicja

Zmienna α występuje w danym miejscu w wyrażeniu ϕ jako zmienna wolna, gdy występuje w tym miejscu, lecz nie bezpośrednio za kwantyfikatorem ani w zasięgu kwantyfikatora. Zmienna α jest wolna w ϕ jeżeli występuje w ϕ w jakimś miejscu jako zmienna wolna.

Definicja

W wyrażeniach $\forall\alpha \phi$ i $\exists\alpha \phi$ kwantyfikator wiąże zmienną α . Jeżeli α nie jest wolne w ϕ , to kwantyfikator wiąże α nieistotnie.

Wolność i niewola zmiennych - przykłady

Tabela: Przykłady wolnych i związań zmiennych w PZW_{WRP}

PZW _{KRZ}	wolne zmienne	związane zmienne
$A(x) \rightarrow B(x)$	x	-
$\forall x \exists y P(x, y, z)$	z	x, y
$\forall x \exists y P(x, y, z) \rightarrow A(x)$	x, z	y

Operacja podstawiania – ogólnie

Definicja

Jeżeli zmienna α jest wolna w wyrażeniu ϕ , to wyrażenie $\phi(\alpha/\xi)$ jest wyrażeniem uzyskanym z ϕ przez prawidłowe podstawienie za α wyrażenia ξ tej samej kategorii składniowej, co α . Przy tym:

- ① W każdym miejscu, w którym α występuje w ϕ jako zmienna wolna, podstawiamy to samo wyrażenie ξ ,
- ② Żadna zmienna wolna ξ nie może stać się związaną w wyniku podstawienia.

Denotacja ξ :

- $x, y, z, \dots,$
- $a, b, c, a_1, b_1, \dots,$
- $a_x, b_{y,z}, c, \dots,$

Operacja podstawiania

Definicja

Jeżeli zmienna α jest wolna w wyrażeniu ϕ , to wyrażenie $\phi(\alpha/\xi)$ jest wyrażeniem uzyskanym z ϕ przez prawidłowe podstawienie za α stałej ξ .

Przy tym:

- 1 W każdym miejscu, w którym α występuje w ϕ jako zmienna wolna, podstawiamy tę samą stałą ξ .

Denotacja ξ :

- $a, b, c, a_1, b_1, \dots,$

Operacja podstawiania - przykłady

Tabela: Przykłady podstawiania w $\mathbb{PZW}_{\text{WRP}}$

$\mathbb{PZW}_{\text{KRZ}}$	podstawienie	wynik podstawienia
$P(\textcolor{red}{x}, y) \rightarrow R(y, \textcolor{red}{x})$	x/y	$P(y, y) \rightarrow R(y, y)$
$P(\textcolor{red}{x}, y) \rightarrow R(y, \textcolor{red}{x})$	x/a	$P(a, y) \rightarrow R(y, a)$
$\forall x P(\textcolor{blue}{x}, y) \rightarrow R(y, \textcolor{red}{x})$	x/b	$\forall x P(x, y) \rightarrow R(y, b)$
$\exists y P(\textcolor{blue}{x}, y)$	x/y	-
$\exists y P(\textcolor{red}{x}, y)$	x/c	$\exists y P(c, y)$

Nowe atomowe tablice analityczne w WRP

- $\text{TO}\forall:$

$$\begin{array}{c} \forall \alpha \ \phi \\ | \\ \phi(\alpha/\xi) \end{array}$$

gdzie: ξ jest dowolną stałą.

- $\text{TO}\exists:$

$$\begin{array}{c} \exists \alpha \ \phi \\ | \\ \phi(\alpha/\xi) \end{array}$$

gdzie: ξ jest dowolną **NOWĄ** stałą, tj. stałą niewystępującą w danej gałęzi.

Nowe atomowe tablice analityczne w WRP (2)

- TON \forall :

$$\neg \forall \alpha \phi$$
$$|$$
$$\neg \phi(\alpha/\xi)$$

gdzie: ξ jest dowolną **NOWĄ** stałą, tj. stałą niewystępującą w danej gałęzi.

- TON \exists :

$$\neg \exists \alpha \phi$$
$$|$$
$$\neg \phi(\alpha/\xi)$$

gdzie: ξ jest dowolną stałą.

Metoda tablic analitycznych - przykłady

Czy wyrażenie $\forall x A(x) \rightarrow \exists x A(x)$ jest prawem logiki?

Metoda tablic analitycznych - przykłady

$$\neg(\forall x A(x) \rightarrow \exists x A(x))$$

Metoda tablic analitycznych - przykłady

$$\begin{array}{c} \neg(\forall x A(x) \rightarrow \exists x A(x)) \\ | \\ \forall x A(x) \\ | \\ \neg\exists x A(x) \end{array}$$

Metoda tablic analitycznych - przykłady

$$\begin{array}{c} \neg(\forall x A(x) \rightarrow \exists x A(x)) \\ | \\ \forall x A(x) \\ | \\ \neg\exists x A(x)) \\ | \\ A(a) \end{array}$$

Metoda tablic analitycznych - przykłady

$$\begin{array}{c} \neg(\forall x A(x) \rightarrow \exists x A(x)) \\ | \\ \forall x A(x) \\ | \\ \neg\exists x A(x)) \\ | \\ A(a) \\ | \\ \neg A(a) \end{array}$$

Metoda tablic analitycznych - przykłady

Tak, wyrażenie $\forall x A(x) \rightarrow \exists x A(x)$ jest prawem logiki.

Metoda tablic analitycznych - przykłady (2)

Czy wyrażenie $\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)$ jest prawem logiki?

Metoda tablic analitycznych - przykłady (2)

$$\neg(\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y))$$

Metoda tablic analitycznych - przykłady (2)

$$\begin{array}{c} \neg(\exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y)) \\ | \\ \exists x \forall y P(x,y) \\ | \\ \neg \forall y \exists x P(x,y) \end{array}$$

Metoda tablic analitycznych - przykłady (2)

$$\neg(\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y))$$
$$\quad \quad \quad \exists x \forall y \underset{|}{P(x, y)}$$
$$\quad \quad \quad \neg \forall y \exists x \underset{|}{P(x, y)}$$
$$\quad \quad \quad \forall y \underset{|}{P(a, y)}$$

Metoda tablic analitycznych - przykłady (2)

$$\neg(\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y))$$

$$\exists x \forall y \underset{|}{P}(x, y)$$

$$\neg \forall y \exists x \underset{|}{P}(x, y)$$

$$\forall y \underset{|}{P}(a, y)$$

$$\neg \exists x \underset{|}{P}(x, b)$$

Metoda tablic analitycznych - przykłady (2)

$$\begin{array}{c} \neg(\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)) \\ | \\ \exists x \forall y P(x, y) \\ | \\ \neg \forall y \exists x P(x, y) \\ | \\ \forall y P(a, y) \\ | \\ \neg \exists x P(x, b) \\ | \\ P(a, b) \end{array}$$

Metoda tablic analitycznych - przykłady (2)

$$\neg(\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y))$$

$$\exists x \forall y \underset{|}{P}(x, y)$$

$$\neg \forall y \exists x \underset{|}{P}(x, y)$$

$$\forall y \underset{|}{P}(a, y)$$

$$\neg \exists x \underset{|}{P}(x, b)$$

$$P(a, b)$$

$$\neg P(a, b)$$

Metoda tablic analitycznych - przykłady (2)

$$\neg(\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y))$$

$$\exists x \forall y \underset{|}{P}(x, y)$$

$$\neg \forall y \exists x \underset{|}{P}(x, y)$$

$$\forall y \underset{|}{P}(a, y)$$

$$\neg \exists x \underset{|}{P}(x, b)$$

$$P(a, b)$$

$$\neg P(a, b)$$

$$P(a, a)$$

$$P(b, b)$$

Metoda tablic analitycznych - przykłady (2)

Tak, wyrażenie $\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)$ jest prawem logiki.

Metoda tablic analitycznych - przykłady (3)

Czy wyrażenie $\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x)$ jest prawem logiki?

Metoda tablic analitycznych - przykłady (3)

$$\neg(\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x))$$

Metoda tablic analitycznych - przykłady (3)

$$\begin{array}{c} \neg(\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x)) \\ | \\ \forall x (A(x) \vee B(x)) \\ | \\ \neg(\forall x A(x) \vee \forall x B(x)) \end{array}$$

Metoda tablic analitycznych - przykłady (3)

$$\neg(\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x))$$

$$\begin{array}{c} \forall x (A(x) \vee B(x)) \\ | \\ \neg(\forall x A(x) \vee \forall x B(x)) \end{array}$$

Metoda tablic analitycznych - przykłady (3)

$$\begin{array}{c} \neg(\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x)) \\ | \\ \forall x (A(x) \vee B(x)) \\ | \\ \neg(\forall x A(x) \vee \forall x B(x)) \\ | \\ \neg\forall x A(x) \\ | \\ \neg\forall x B(x) \\ | \\ \neg A(a) \end{array}$$

Metoda tablic analitycznych - przykłady (3)

$$\begin{array}{c} \neg(\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x)) \\ | \\ \forall x (A(x) \vee B(x)) \\ | \\ \neg(\forall x A(x) \vee \forall x B(x)) \\ | \\ \neg\forall x A(x) \\ | \\ \neg\forall x B(x) \\ | \\ \neg A(a) \\ | \\ \neg B(b) \end{array}$$

Metoda tablicy analitycznych - przykłady (3)

$$\neg(\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x))$$

$$\begin{array}{c} \forall x (A(x) \vee B(x)) \\ | \\ \forall x (A(x) \vee B(x)) \end{array}$$

$$\begin{array}{c} \neg(\forall x A(x) \vee \forall x A(x)) \\ | \\ \neg\forall x A(x) \end{array}$$

$$\begin{array}{c} \neg\forall x A(x) \\ | \\ \neg\forall x B(x) \end{array}$$

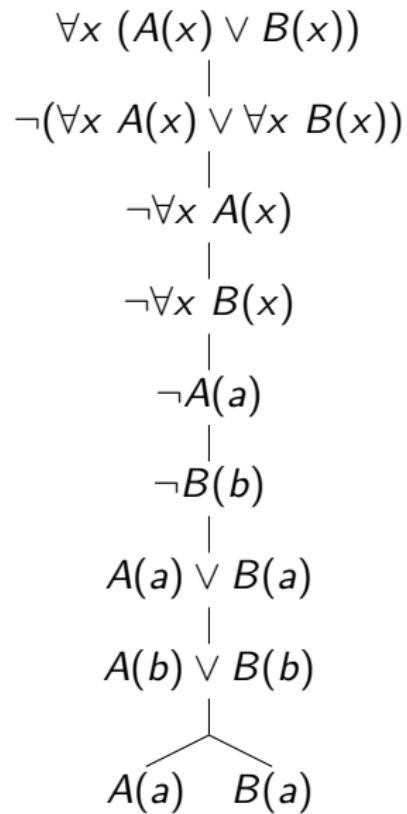
$$\begin{array}{c} \neg\forall x B(x) \\ | \\ \neg A(a) \end{array}$$

$$\begin{array}{c} \neg A(a) \\ | \\ \neg B(b) \end{array}$$

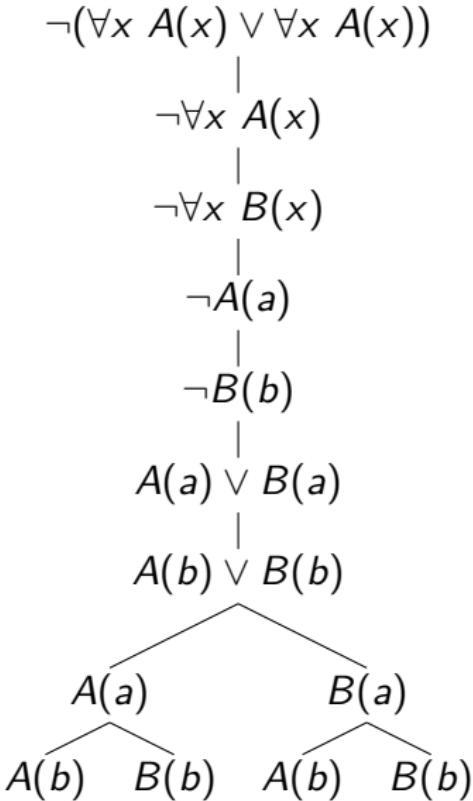
$$\begin{array}{c} \neg B(b) \\ | \\ A(a) \vee B(a) \end{array}$$

$$\begin{array}{c} A(a) \vee B(a) \\ | \\ A(b) \vee B(b) \end{array}$$

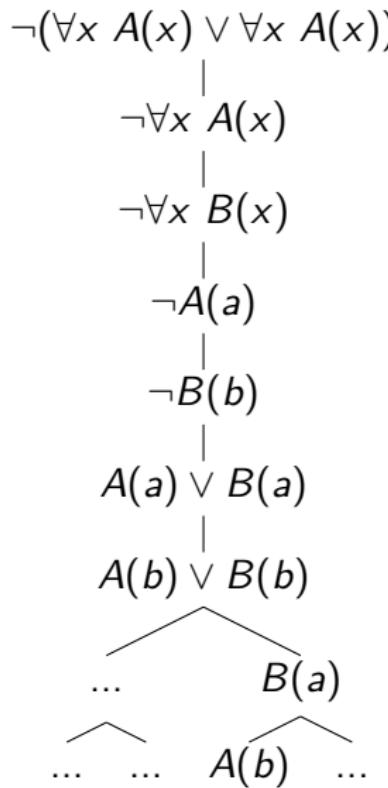
Metoda tablicy analitycznych - przykłady (3)



Metoda tablicy analitycznych - przykłady (3)



Metoda tablic analitycznych - przykłady (3)



Metoda tablic analitycznych - przykłady (3)

Nie, wyrażenie $\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x A(x)$ **nie** jest prawem logiki.

Metoda tablic analitycznych - przykłady (4)

Czy wyrażenie $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$ jest prawem logiki?

Metoda tablic analitycznych - przykłady (4)

$$\neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y))$$

Metoda tablic analitycznych - przykłady (4)

$$\begin{array}{c} \neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)) \\ | \\ \forall y \exists x P(x, y) \\ | \\ \neg \exists x \forall y P(x, y) \end{array}$$

Metoda tablic analitycznych - przykłady (4)

$$\begin{array}{c} \neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)) \\ | \\ \forall y \exists x P(x, y) \\ | \\ \neg \exists x \forall y P(x, y) \\ | \\ \exists x P(x, a) \end{array}$$

Metoda tablic analitycznych - przykłady (4)

$$\neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y))$$

$$\begin{array}{c} \forall y \exists x \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \neg \exists x \forall y \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \exists x \\ | \\ P(x, a) \end{array}$$

$$\begin{array}{c} | \\ P(b, a) \end{array}$$

Metoda tablic analitycznych - przykłady (4)

$$\neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y))$$

$$\forall y \exists x \underset{|}{P}(x, y)$$

$$\neg \exists x \forall y \underset{|}{P}(x, y)$$

$$\exists x \underset{|}{P}(x, a)$$

$$P(b, a)$$

$$\neg \forall y P(a, y)$$

Metoda tablicy analitycznych - przykłady (4)

$$\neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y))$$

$$\begin{array}{c} \forall y \exists x \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \neg \exists x \forall y \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \exists x \\ | \\ P(x, a) \end{array}$$

$$\begin{array}{c} | \\ P(b, a) \end{array}$$

$$\begin{array}{c} | \\ \neg \forall y \\ | \\ P(a, y) \end{array}$$

$$\begin{array}{c} | \\ \neg \forall y \\ | \\ P(b, y) \end{array}$$

Metoda tablicy analitycznych - przykłady (4)

$$\neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y))$$

$$\begin{array}{c} \forall y \exists x \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \neg \exists x \forall y \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \exists x \\ | \\ P(x, a) \end{array}$$

$$\begin{array}{c} | \\ P(b, a) \end{array}$$

$$\begin{array}{c} | \\ \neg \forall y P(a, y) \end{array}$$

$$\begin{array}{c} | \\ \neg \forall y P(b, y) \end{array}$$

$$\begin{array}{c} | \\ \neg P(a, c) \end{array}$$

Metoda tablicy analitycznych - przykłady (4)

$$\neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y))$$

$$\begin{array}{c} \forall y \exists x \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \neg \exists x \forall y \\ | \\ P(x, y) \end{array}$$

$$\begin{array}{c} \exists x \\ | \\ P(x, a) \end{array}$$

$$\begin{array}{c} | \\ P(b, a) \end{array}$$

$$\begin{array}{c} | \\ \neg \forall y \\ | \\ P(a, y) \end{array}$$

$$\begin{array}{c} | \\ \neg \forall y \\ | \\ P(b, y) \end{array}$$

$$\begin{array}{c} | \\ \neg P(a, c) \end{array}$$

$$\begin{array}{c} | \\ \neg P(b, d) \end{array}$$

Metoda tablicy analitycznych - przykłady (4)

$$\neg(\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y))$$

$$\begin{array}{c} \forall y \exists x P(x, y) \\ | \\ \neg \exists x \forall y P(x, y) \\ | \\ \exists x P(x, a) \\ | \\ P(b, a) \\ | \\ \neg \forall y P(a, y) \\ | \\ \neg \forall y P(b, y) \\ | \\ \neg P(a, c) \\ | \\ \neg P(b, d) \\ | \\ \exists x P(x, d) \end{array}$$

Metoda tablic analitycznych - przykłady (4)

$$\begin{array}{c} \forall y \exists x P(x, y) \\ | \\ \neg \exists x \forall y P(x, y) \\ | \\ \exists x P(x, a) \\ | \\ P(b, a) \\ | \\ \neg \forall y P(a, y) \\ | \\ \neg \forall y P(b, y) \\ | \\ \neg P(a, c) \\ | \\ \neg P(b, d) \\ | \\ \exists x P(x, d) \\ | \\ P(e, d) \end{array}$$

Metoda tablic analitycznych - przykłady (4)

- Metoda tablic analitycznych nie jest w stanie rozstrzygnąć, czy $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$ jest prawem logiki.

Metoda tablic analitycznych - przykłady (4)

- Metoda tablic analitycznych nie jest w stanie rozstrzygnąć, czy $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$ jest prawem logiki.
 - ▶ W WRP metoda tablic analitycznych jest metodą półalgoritmiczną.

Metoda tablic analitycznych - przykłady (4)

- Metoda tablic analitycznych nie jest w stanie rozstrzygnąć, czy $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$ jest prawem logiki.
 - ▶ W WRP metoda tablic analitycznych jest metodą półalgoritmiczną.
 - ▶ Nie istnieje algorytmiczna metoda szukania praw WRP - WRP jest teorią nierożstrzygalną.

Metoda tablic analitycznych - przykłady (4)

- Wyrażenie $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$ **nie** jest prawem logiki.

Metoda tablic analitycznych - przykłady (4)

- Wyrażenie $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$ **nie** jest prawem logiki.
 - ▶ $P(x, y) - x > y.$

Ograniczenia metody tablic analitycznych dla WRP

- półalgotytmiczność:
 - ▶ niektóre tablice analityczne dla nie-praw WRP są nieskończone
- czasami trzeba “oszukiwać”:
 - ▶ tablica dla $A(x) \rightarrow \exists x A(x)$ nie jest zamknięta, mimo że jest to prawo WRP,
 - ▶ jednak wyrażenie $A(x) \rightarrow \exists x A(x)$ jest równoważne $\forall y (A(y) \rightarrow \exists x A(x))$, dla którego tablica analityczna jest zamknięta.

Sprawdzanie poprawnosci wnioskowań

Żaden gryzoń nie jest gadem.

Nieprawda, że niektóre gady są gryzoniami.

Dłużnicy naszych dłużników są naszymi dłużnikami.

Każdy jest dłużnikiem swego każdego wierzyciela.

Każdy jest wierzycielem swego każdego dłużnika.

Wierzyciele naszych wierzycieli są naszymi wierzycielami.

Sprawdzanie poprawności wnioskowań (2)

Żaden gryzoń nie jest gadem.

Nieprawda, że niektóre gady są gryzoniami.

- $A(x)$ – x jest gryzoniem
- $B(x)$ – x jest gadem

$$\forall x(A(x) \rightarrow \neg B(x))$$

$$\neg \exists x(B(x) \wedge A(x))$$

$$\forall x(A(x) \rightarrow \neg B(x)) \rightarrow \neg \exists x(B(x) \wedge A(x))$$

Sprawdzanie poprawnosci wnioskowań (3)

Dłużnicy naszych dłużników są naszymi dłużnikami.

Każdy jest dłużnikiem swego każdego wierzyciela.

Każdy jest wierzycielem swego każdego dłużnika.

Wierzyciele naszych wierzycieli są naszymi wierzycielami.

- $P(x, y) - x$ jest dłużnikiem y
- $Q(x, y) - x$ jest wierzycielem y

$$\forall x \forall y \forall z [P(y, z) \wedge P(x, y) \rightarrow P(x, z)]$$

$$\forall x \forall y [Q(x, y) \rightarrow P(y, x)]$$

$$\forall x \forall y [P(x, y) \rightarrow Q(y, x)]$$

$$\forall x \forall y \forall z [Q(y, z) \wedge Q(x, y) \rightarrow Q(x, z)]$$

Modele WRP

Definicja

Parę $\langle \mathbb{D}, \mathbb{I} \rangle$ nazywamy modelem dla języka WRP jeżeli:

- ① $\mathbb{D} \neq \emptyset$,
- ② \mathbb{I} jest taką funkcją, która:

- ③ odwzorowuje zbiór CONST stałych języka WRP w zbiór \mathbb{D} ,
- ④ odwzorowuje zbiór predykatów jednoargumentowych REL w $\wp(\mathbb{D})$ (=zbiór wszystkich podzbiorów \mathbb{D}),
- ⑤ odwzorowuje zbiór predykatów dwuargumentowych REL w $\wp(\mathbb{D} \times \mathbb{D})$ (=zbiór wszystkich podzbiorów iloczynu kartezjańskiego \mathbb{D} "razy" \mathbb{D})
- ⑥ ...,
- ⑦ odwzorowuje zbiór predykatów n -argumentowych REL w $\wp(\mathbb{D}^n)$ (=zbiór wszystkich podzbiorów n -argumentowego iloczynu kartezjańskiego $\underbrace{\mathbb{D} \times \mathbb{D} \times \dots \times \mathbb{D}}_{n \text{ razy}}$).

Zbiór \mathbb{D} nazywamy dziedziną modelu.

Wartościowanie w WRP

Definicja

Niech $\mathfrak{M} = \langle \mathbb{D}, \mathbb{I} \rangle$ będzie modelem WRP.

- Niech $x \in \mathbb{D}$.
- Jeżeli \mathfrak{v} jest wartościowaniem, to $\mathfrak{v}[\alpha/x]$ jest takim wartościowaniem, że:

$$\mathfrak{v}[\alpha/x](\beta) = \begin{cases} \mathfrak{v}(\beta) & \text{jeżeli } \alpha \neq \beta \\ x & \text{jeżeli } \alpha = \beta. \end{cases} \quad (19)$$

Spełnianie w WRP

Niech $\mathfrak{M} = \langle \mathbb{D}, \mathbb{I} \rangle$ będzie modelem WRP.

- $\mathfrak{M}, v \models \phi$, to tyle, co ϕ jest spełnione w modelu \mathfrak{M} przy wartościowaniu v .
- $\mathfrak{M}, v \not\models \phi$ to tyle, co ϕ **nie** jest spełnione w modelu \mathfrak{M} przy wartościowaniu v .
- $\mathfrak{M} \models \phi$, to tyle, co ϕ jest spełnione w modelu \mathfrak{M} .
- $\models \phi$, to tyle, co ϕ jest tautologią.

Spełnianie w WRP (2)

$$\mathfrak{M}, \mathfrak{v} \models \delta_n^{(s_n)}(\alpha_1, \alpha_2, \dots, \alpha_{s_n}) \equiv <\mathfrak{v}(\alpha_1), \mathfrak{v}(\alpha_2), \dots, \mathfrak{v}(\alpha_{s_n})> \in \mathbb{I}(\delta_n^{(s_n)}) \quad (20)$$

$$\mathfrak{M}, \mathfrak{v} \models \neg\phi \equiv \mathfrak{M} \not\models \phi \quad (21)$$

$$\mathfrak{M}, \mathfrak{v} \models \phi \wedge \psi \equiv \mathfrak{M} \models \phi \text{ oraz } \mathfrak{M} \models \psi \quad (22)$$

itd.

$$\mathfrak{M}, \mathfrak{v} \models \forall \alpha \phi \equiv \text{ dla dowolnego } x \in \mathbb{D} \mathfrak{M}, \mathfrak{v}[\alpha/x] \models \phi \quad (23)$$

$$\mathfrak{M}, \mathfrak{v} \models \exists \alpha \phi \equiv \text{ dla jakiegoś } x \in \mathbb{D} \mathfrak{M}, \mathfrak{v}[\alpha/x] \models \phi \quad (24)$$

Spełnianie w WRP (3)

$$\mathfrak{M} \models \phi \equiv \forall \mathfrak{v} \ \mathfrak{M}, \mathfrak{v} \models \phi. \quad (25)$$

$$\models \phi \equiv \forall \mathfrak{M} \ \mathfrak{M} \models \phi. \quad (26)$$

Spełnianie w WRP a tablice analityczne

- Jeżeli $\models \phi$, to istnieje skończona zamknięta tablica analityczna dla ϕ .
- Jeżeli $\neg \models \phi$, to istnieje, skończona lub nieskończona, otwarta tablica analityczna dla ϕ .

Semantyka dla języka PRZYKŁAD

PRZYKŁAD – Alfabet

$\langle \text{const} \rangle ::= "Donald" | "Jarosław"$ (27)

$\langle \text{var} \rangle ::= "x" | "y" | "z"$ (28)

$\langle \text{nominal} \rangle ::= \langle \text{const} \rangle \mid \langle \text{var} \rangle$ (29)

$\langle \text{seq2} \rangle ::= \langle \text{nominal} \rangle , " , " \langle \text{nominal} \rangle$ (30)

$\langle \text{pred} \rangle ::= "jest przyjacielem"$ (31)

PRZYKŁAD – Język

$<atom> ::= <pred> "(" <seq2> ")"$ (32)

$<\phi> ::= <atom> |$

" \forall " $<var>$ ϕ |

" \exists " $<var>$ ϕ |

" \neg " $<\phi>$ |

$<\phi>$ " \wedge " $<\phi>$ |

$<\phi>$ " \vee " $<\phi>$ |

$<\phi>$ " \rightarrow " $<\phi>$ |

$<\phi>$ " \equiv " $<\phi>$

(33)

PRZYKŁAD – Model \mathfrak{M}_1

- ① $\mathbb{D} = \{\odot, \ominus, \oplus\}$
- ② $\mathbb{D} \times \mathbb{D}$
- ③ $\mathbb{I}(\text{"Donald"}) = \odot$
- ④ $\mathbb{I}(\text{"Jarosław"}) = \ominus$
- ⑤ $\mathbb{I}(\text{"jest przyjacielem"}) = \{<\odot, \ominus>, <\odot, \oplus>\}$

PRZYKŁAD – Wartościowanie v_1

- $v_1("x") = \odot$
- $v_1("y") = \odot$
- $v_1("z") = \odot$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_1

- $\mathfrak{M}_1, \mathfrak{v}_1 \models "Donald \text{ jest przyjacielem Jarosław}"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "Donald \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \not\models "Jarosław \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \not\models "Jarosław \text{ jest przyjacielem Jarosław}"$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_1 (2)

- $\mathfrak{M}_1, \mathfrak{v}_1 \models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \not\models "x \text{ jest przyjacielem } y"$

PRZYKŁAD – Wartościowanie v_2

- $v_1("x") = \odot$
- $v_1("y") = \ominus$
- $v_1("z") = \bullet$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_2

- $\mathfrak{M}_1, \mathfrak{v}_2 \models "Donald \text{ jest przyjacielem Jarosław}"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "Donald \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \not\models "Jarosław \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \not\models "Jarosław \text{ jest przyjacielem Jarosław}"$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_2 (2)

- $\mathfrak{M}_1, \mathfrak{v}_2 \models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \not\models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "x \text{ jest przyjacielem } y"$

PRZYKŁAD – Model \mathfrak{M}_2

- ① $\mathbb{D} = \{\odot, \oplus, \ominus\}$
- ② $\mathbb{I}(\text{"Donald"}) = \odot$
- ③ $\mathbb{I}(\text{"Jarosław"}) = \oplus$
- ④ $\mathbb{I}(\text{"jest przyjacielem"}) = \mathbb{D} \times \mathbb{D}$

PRZYKŁAD – Spełnianie w \mathfrak{M}_2 i dla v_1

- $\mathfrak{M}_2, v_1 \models "Donald \text{ jest przyjacielem Jarosław"}$
- $\mathfrak{M}_2, v_1 \models "Donald \text{ jest przyjacielem Donald"}$
- $\mathfrak{M}_2, v_1 \models "Jarosław \text{ jest przyjacielem Donald"}$
- $\mathfrak{M}_2, v_1 \models "Jarosław \text{ jest przyjacielem Jarosław"}$

PRZYKŁAD – Spełnianie w \mathfrak{M}_2 i dla \mathfrak{v}_1 (2)

- $\mathfrak{M}_2, \mathfrak{v}_1 \models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "x \text{ jest przyjacielem } y"$

PRZYKŁAD – Model \mathfrak{M}_3

- ① $\mathbb{D} = \{\odot, \oplus, \ominus\}$
- ② $\mathbb{I}(\text{"Donald"}) = \odot$
- ③ $\mathbb{I}(\text{"Jarosław"}) = \oplus$
- ④ $\mathbb{I}(\text{"jest przyjacielem"}) = \emptyset$

PRZYKŁAD – Spełnianie w \mathfrak{M}_3 i dla \mathfrak{v}_2

- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Donald\ jest\ przyjacielem\ Jarosław"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Donald\ jest\ przyjacielem\ Donald"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Jarosław\ jest\ przyjacielem\ Donald"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Jarosław\ jest\ przyjacielem\ Jarosław"$

PRZYKŁAD – Spełnianie w \mathfrak{M}_3 i dla \mathfrak{v}_2 (2)

- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "x \text{ jest przyjacielem } y"$

Modele

PRZYKŁAD – Alfabet

$\langle \text{const} \rangle ::= "Donald" \text{ Jarosław}$ (34)

$\langle \text{var} \rangle ::= "x" y" z"$ (35)

$\langle \text{nominal} \rangle ::= \langle \text{const} \rangle \mid \langle \text{var} \rangle$ (36)

$\langle \text{seq2} \rangle ::= \langle \text{nominal} \rangle ", \text{nominal} \rangle$ (37)

$\langle \text{pred} \rangle ::= "jest przyjacielem"$ (38)

PRZYKŁAD – Język

$< atom > ::= < pred > "(seq2 > ")"$ (39)

$< \phi > ::= < atom > |$

$"\forall var > \phi |$

$"\exists var > \phi |$

$"\neg \phi > |$

$< \phi > " \wedge \phi > |$

$< \phi > " \vee \phi > |$

$< \phi > " \rightarrow \phi > |$

$< \phi > " \equiv \phi >$ (40)

PRZYKŁAD – Model \mathfrak{M}_1

- ① $\mathbb{D} = \{\odot, \oplus, \ominus\}$
- ② $\mathbb{I}(\text{"Donald"}) = \odot$
- ③ $\mathbb{I}(\text{"Jarosław"}) = \oplus$
- ④ $\mathbb{I}(\text{"jest przyjacielem"}) = \{<\odot, \oplus>, <\oplus, \ominus>\}$

PRZYKŁAD – Wartościowanie v_1

- $v_1("x") = \odot$
- $v_1("y") = \odot$
- $v_1("z") = \odot$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_1

- $\mathfrak{M}_1, \mathfrak{v}_1 \models "Donald \text{ jest przyjacielem Jarosław}"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "Donald \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \not\models "Jarosław \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \not\models "Jarosław \text{ jest przyjacielem Jarosław}"$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_1 (2)

- $\mathfrak{M}_1, \mathfrak{v}_1 \models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_1, \mathfrak{v}_1 \not\models "x \text{ jest przyjacielem } y"$

PRZYKŁAD – Wartościowanie v_2

- $v_1("x") = \odot$
- $v_1("y") = \ominus$
- $v_1("z") = \bullet$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_2

- $\mathfrak{M}_1, \mathfrak{v}_2 \models "Donald \text{ jest przyjacielem Jarosław}"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "Donald \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \not\models "Jarosław \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \not\models "Jarosław \text{ jest przyjacielem Jarosław}"$

PRZYKŁAD – Spełnianie w \mathfrak{M}_1 i dla \mathfrak{v}_2 (2)

- $\mathfrak{M}_1, \mathfrak{v}_2 \models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \not\models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_1, \mathfrak{v}_2 \models "x \text{ jest przyjacielem } y"$

PRZYKŁAD – Model \mathfrak{M}_2

- ① $\mathbb{D} = \{\odot, \oplus, \ominus\}$
- ② $\mathbb{I}(\text{"Donald"}) = \odot$
- ③ $\mathbb{I}(\text{"Jarosław"}) = \oplus$
- ④ $\mathbb{I}(\text{"jest przyjacielem"}) = \mathbb{D} \times \mathbb{D}$

PRZYKŁAD – Spełnianie w \mathfrak{M}_2 i dla v_1

- $\mathfrak{M}_2, v_1 \models "Donald \text{ jest przyjacielem Jarosław"}$
- $\mathfrak{M}_2, v_1 \models "Donald \text{ jest przyjacielem Donald"}$
- $\mathfrak{M}_2, v_1 \models "Jarosław \text{ jest przyjacielem Donald"}$
- $\mathfrak{M}_2, v_1 \models "Jarosław \text{ jest przyjacielem Jarosław"}$

PRZYKŁAD – Spełnianie w \mathfrak{M}_2 i dla \mathfrak{v}_1 (2)

- $\mathfrak{M}_2, \mathfrak{v}_1 \models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_2, \mathfrak{v}_1 \models "x \text{ jest przyjacielem } y"$

PRZYKŁAD – Model \mathfrak{M}_3

- ① $\mathbb{D} = \{\odot, \oplus, \ominus\}$
- ② $\mathbb{I}(\text{"Donald"}) = \odot$
- ③ $\mathbb{I}(\text{"Jarosław"}) = \oplus$
- ④ $\mathbb{I}(\text{"jest przyjacielem"}) = \emptyset$

PRZYKŁAD – Spełnianie w \mathfrak{M}_3 i dla \mathfrak{v}_2

- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Donald \text{ jest przyjacielem Jarosław}"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Donald \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Jarosław \text{ jest przyjacielem Donald}"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "Jarosław \text{ jest przyjacielem Jarosław}"$

PRZYKŁAD – Spełnianie w \mathfrak{M}_3 i dla \mathfrak{v}_2 (2)

- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "x \text{ jest przyjacielem } x"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "y \text{ jest przyjacielem } y"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "\exists x \exists y \ x \text{ jest przyjacielem } y"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \models "\exists x \exists y (\ x \text{ jest przyjacielem } y \vee \neg x \text{ jest przyjacielem } y)"$
- $\mathfrak{M}_3, \mathfrak{v}_2 \not\models "x \text{ jest przyjacielem } y"$

Metalogiczne własności WRP

- ① WRP jest niesprzeczny
- ② WRP jest zupełny
- ③ WRP nie jest rozstrzygalny

Prover9/Mace4

- Prover9 – system automatycznego dowodzenia twierdzeń (*prover*)
- Mace4 – system automatycznego wyszukiwania modeli (*model finder*)
- <https://www.cs.unm.edu/~mccune/mace4/>

Język WRP w Prover9/Mace4

<https://www.cs.unm.edu/~mccune/mace4/manual/2009-11A/>

$< zero_digit > ::= "0"$ (41)

$< nonzero_digit > ::= "1" "2" "4" "5" "6" "7" "8" "9"$ (42)

$< digit > ::= < zero_digit > | < nonzero_digit >$ (43)

$< index > ::= < digit > | < nonzero_digit > < index >$ (44)

$< const > ::= ("a" b . . . | "t") [< index >]$ (45)

$< var > ::= ("u" v" x" y" z") [< index >]$ (46)

$< \alpha > ::= < const > | < var >$ (47)

$< seq > ::= < \alpha > | < \alpha > ", seq >$ (48)

$< pred > ::= "A" B . . . | "Z") [< index >]$ (49)

Język WRP w Prover9/Mace4 (2)

$< atom > ::= < pred > " (seq > ") .$ (50)

$< \phi > ::= < atom > |$

$" allvar > \phi .$

$" existsvar > \phi .$

$\phi > .$

$< \phi > "\&\phi > .$

$< \phi > \phi > .$

$< \phi >> \phi > .$

$< \phi > - > \phi > .$ (51)

Dowodzenie w Prover9/Mace4

- ① $\forall x A(x) \rightarrow \exists x A(x)$
- ② $A(x) \rightarrow \exists x A(x)$
- ③ $\exists x \forall y P(x, y) \rightarrow \forall y \exists x P(x, y)$
- ④ $\forall x A(x) \vee \forall x B(x) \rightarrow \forall x (A(x) \vee B(x))$
- ⑤

$$\forall x \forall y \forall z [P(y, z) \wedge P(x, y) \rightarrow P(x, z)]$$

$$\forall x \forall y [Q(x, y) \rightarrow P(y, x)]$$

$$\forall x \forall y [P(x, y) \rightarrow Q(y, x)]$$

$$\forall x \forall y \forall z [Q(y, z) \wedge Q(x, y) \rightarrow Q(x, z)]$$

Szukanie modeli w Prover9/Mace4

- ① $\forall y \exists x P(x, y) \rightarrow \exists x \forall y P(x, y)$
- ② $\forall x (A(x) \vee B(x)) \rightarrow \forall x A(x) \vee \forall x B(x)$

Inne systemy automatycznego dowodzenia twierdzeń

<http://tptp.cs.miami.edu/>

Spis treści

1 Klasyczny rachunek zdań

- Rzut oka wstecz
- Metoda tablic analitycznych
- Metalogiczne własności KRZ

2 Węższy rachunek predykatów

- Motywacja
- Język WRP
- Metoda tablic analitycznych
- Semantyka dla WRP
- Metalogiczne własności WRP
- Systemy automatycznego dowodzenia twierdzeń

3 Logiki opisowe

- Logika opisowa \mathcal{ALC}
- Reprezentacja relacji w logice
- Logika opisowa \mathcal{SROIQ}

Logiki opisowe

- *Description Logics - [?]*
- geneza:
 - ▶ koniec lat 80-ych XX w.
 - ▶ sieci semantyczne – reprezentacja struktury wypowiedzi w grafach
- rodzina logik
- rozstrzygalne fragmenty WRP
 - ▶ czasami (optymalnie): liczba kroków w rezolucji wyrażenia zależy wykładniczo od liczby predykatów w wyrażeniu
- “brak” aksjomatów
- zastosowania: Sieć Semantyczna

Sieć Semantyczna a ontologia inżynierjna

Sieć Semantyczna nie jest osobną siecią, lecz rozszerzeniem obecnej, w którym informacja posiada dobrze zdefiniowane znaczenie, co zwiększa możliwości komputerów i ludzi do współpracy. [...] maszyny będą lepiej przetwarzać i „rozumieć” te dane, które obecnie tylko wyświetlają. [...] Aby Sieć Semantyczna mogła funkcjonować, komputery muszą mieć dostęp do ustrukturyzowanych zasobów danych oraz zbiorów reguł wnioskowania, których będą mogły użyć do procesów automatycznego wnioskowania.

Założenia konstrukcyjne logik opisowych

- Logika opisowa konstruuje pojęcia (*concepts*) i za pomocą tych pojęć opisuje przedmioty indywidualne.
- Pojęcia są budowane z:
 - ① pojęć atomicznych (predykatów jednoargumentowych)
 - ② ról (predykatów binarnych)
- Typ logiki opisowej zależy od zbioru dostępnych sposobów konstrukcji pojęć złożonych z pojęć atomicznych i ról.
- Wiedza dotycząca pojęć i indywidualów może być wyprowadzona automatycznie. W szczególności, możemy “dowiadywać się” o relacji subsumpcji pomiędzy pojęciami.

Pudełka semantyczne

① TBox - opis terminologii

- ▶ Każdy człowiek jest zwierzęciem.
- ▶ Każde zwierzę ma duszę.
- ▶ Dusza każdego człowieka jest niematerialna.

② ABox - opis świata

- ▶ Jan jest człowiekiem.
- ▶ Burek jest zwierzęciem, ale nie jest człowiekiem.
- ▶ Warszawa nie jest zwierzęciem.

Logika opisowa \mathcal{ALC}

- język
- model
- tablice semantyczne

Alfabet \mathcal{ALC}

Definicja

Alfabet \mathcal{ALC} zawiera:

- ① zbiór stałych indywidualnych $\text{CONST} := \{a_1, a_2, \dots\}$,
- ② zbiór predykatów

▶ jednoargumentowych (pojęć atomicznych):

$$\text{CONCEPT} := \{\top, A_1, A_2, \dots\}$$

▶ dwuargumentowych (ról):

$$\text{ROLE} := \{R_1, R_2, \dots\}$$

- ③ funktry (nazwotwórcze od argumentów nazwowych): $-$, \sqcap ;
- ④ "kwantyfikatory": $\exists.$, $\forall.$;
- ⑤ funktry (zdaniotwórcze od argumentów nazwowych): \sqsubseteq , $=$;
- ⑥ nawiasy: $(,)$.

“Kwantyfikatory” w logikach opisowych

- $\exists R.C$ - (predykat reprezentujący) zbiór takich przedmiotów, które są powiązane relacją R z *przynajmniej jednym* przedmiotem ze zbioru C

$$\exists R.C(x) \equiv \exists y[R(x, y) \wedge C(y)]$$

- $\forall R.C$ - (predykat reprezentujący) zbiór takich przedmiotów, które są powiązane relacją R *tylko* z przedmiotami ze zbioru C

$$\forall R.C(x) \equiv \forall y[R(x, y) \rightarrow C(y)]$$

- przykłady:

- ▶ R - jest potomkiem, C - jest Polakiem
 - ★ $\exists R.C$ - potomkowie (jakichś) Polaków
 - ★ $\forall R.C$ - prawdziwi Polacy (potomkowie tylko Polaków)
- ▶ R - zna, C - jest celebrytą
 - ★ $\exists R.C$ - ci, którzy znają jakiegoś celebrytę
 - ★ $\forall R.C$ - ci, którzy znają tylko celebrytów

Zbiór pojęć logiki \mathcal{ALC}

Dowolny ciąg elementów alfabetu \mathcal{ALC} jest *napisem* logiki \mathcal{ALC} .

Definicja

Zbiór pojęć logiki \mathcal{ALC} jest najmniejszym zbiorem napisów \mathcal{ALC} spełniającym poniższe warunki:

- ① Każde pojęcie atomiczne jest pojęciem \mathcal{ALC} .
- ② Jeżeli δ jest pojęciem, to $-\delta$ jest pojęciem.
- ③ Jeżeli δ_1, δ_2 są pojęciami, to $\delta_1 \sqcap \delta_2$ jest pojęciem.
- ④ Jeżeli ρ jest rolą i δ jest pojęciem, to $\exists\rho.\delta$ oraz $\forall\rho.\delta$ są pojęciami.

Zbiór pojęć logiki \mathcal{ALC} - przykłady

HumanBeing, Female, Polish, \top

Female \sqcap Polish

\neg HumanBeing

Female \sqcap HumanBeing

\neg \top

\neg (Female \sqcap Polish)

Zbiór pojęć logiki \mathcal{ALC} - przykłady (2)

$\exists hasChild. \top$

$\forall hasHair. \bot$

$Female \sqcap (\exists hasChild. \top)$

$(\neg Female) \sqcap HumanBeing \sqcap (\exists hasChild. \top) \sqcap \forall hasHair. \bot$

Język logiki \mathcal{ALC}

Definicja

Zbiór poprawnie zbudowanych wyrażeń logiki \mathcal{ALC} jest najmniejszym zbiorem napisów \mathcal{ALC} spełniającym warunek:

- ① Jeśli δ_1, δ_2 są pojęciami logiki \mathcal{ALC} , to $\delta_1 \sqsubseteq \delta_2$ oraz $\delta_1 = \delta_2$ są poprawnie zbudowanymi wyrażeniami \mathcal{ALC} .
- ② Jeśli ρ_1, ρ_2 są rolami logiki \mathcal{ALC} , to $\rho_1 \sqsubseteq \rho_2$ oraz $\rho_1 = \rho_2$ są poprawnie zbudowanymi wyrażeniami \mathcal{ALC} .
- ③ Jeśli δ jest pojęciem a α stałą, logiki \mathcal{ALC} , to $\delta(\alpha)$ jest poprawnie zbudowanym wyrażeniem \mathcal{ALC} .
- ④ Jeśli ρ rola, a α_1 i α_2 , stałymi logiki \mathcal{ALC} , to $\rho(\alpha_1, \alpha_2)$ jest poprawnie zbudowanym wyrażeniem \mathcal{ALC} .

Zbiór wyrażeń spełniających warunki 1 lub 2 jest nazywany pudełkiem terminologicznym.

Zbiór wyrażeń spełniających warunki 3 lub 4 jest nazywany pudełkiem asercji.

Język logiki \mathcal{ALC} - przykłady (2)

$Male \sqsubseteq Animal$

$Female \sqsubseteq \neg Animal$

$Man = HumanBeing \sqcap \neg Female$

$Father = Man \sqcap \exists hasChild. \top$

$Bachelor = Man \sqcap \forall hasSpouse. \perp$

$\exists hasChild. \top \sqsubseteq \exists hasSpouse. \top$

$GenuineMan = Man \sqcap (\exists build.House) \sqcap (\exists hasChild.Male) \sqcap (\exists plant.Tree)$

Język logiki \mathcal{ALC} - przykłady (2)

...

*Podobno każdy ma swój kawałek cienia -
brak cienia jest dowodem nieistnienia -
lecz bez kawałka światła
nie jest łatwo ... (G. Turnau, Kawałek cienia)*

$$\top \sqsubseteq \exists \textit{hasShadow}.\top$$

(zainspirowany przez J. Pogonowskiego)

Język logiki \mathcal{ALC} - przykłady (3)

$hasChild \sqsubseteq hasDescendant$

$hasWife = hasHusband$

Język logiki \mathcal{ALC} - przykłady (4)

$HumanBeing(Jan)$

$\top(Jan)$

$(Female \sqcap Polish)(Jan)$

$(\neg HumanBeing)(Burek)$

$(\exists hasChild.\top)(Jan)$

$(\forall hasHair. \neg \top)(Jan)$

$(Female \sqcap (\exists hasChild.\top))(Anna)$

$((\neg Female) \sqcap HumanBeing \sqcap (\exists hasChild.\top) \sqcap \forall hasHair. \neg \top)(Jan)$

Czego nie można powiedzieć w \mathcal{ALC} ?

- Nie istnieje największa liczba pierwsza.
- Każdy kij ma dwa końce.
- Przyjaciele moich przyjaciół są moimi przyjaciółmi.

Modele \mathcal{ALC}

Definicja

Parę $\mathfrak{M} = \langle \mathbb{D}, \mathbb{I} \rangle$ nazywamy modelem dla języka \mathcal{ALC} jeżeli:

- ① $\mathbb{D} \neq \emptyset$,
- ② \mathbb{I} jest taką funkcją, która:

① odwzorowuje zbiór CONST w zbiór \mathbb{D} ,

② odwzorowuje zbiór pojęć w $\wp(\mathbb{D})$, przy czym spełnione są następujące warunki:

$$\mathbb{I}(\top) = \mathbb{D}$$

$$\mathbb{I}(-\delta) = \mathbb{D} \setminus \mathbb{I}(\delta)$$

$$\mathbb{I}(\delta_1 \sqcap \delta_2) = \mathbb{I}(\delta_1) \cap \mathbb{I}(\delta_2)$$

$$\mathbb{I}(\exists \rho. \delta) = \{x \in \mathbb{D} : \exists y < x, y > \in \mathbb{I}(\rho) \wedge y \in \mathbb{I}(\delta)\}$$

$$\mathbb{I}(\forall \rho. \delta) = \{x \in \mathbb{D} : \forall y [< x, y > \in \mathbb{I}(\rho) \rightarrow y \in \mathbb{I}(\delta)]\}$$

Spełnianie w \mathcal{ALC}

$$\mathfrak{M} \models \delta(\alpha) \equiv \mathbb{I}(\alpha) \in \mathbb{I}(\delta) \quad (52)$$

$$\mathfrak{M} \models \rho(\alpha_1, \alpha_2) \equiv <\mathbb{I}(\alpha_1), \mathbb{I}(\alpha_2)> \in \mathbb{I}(\rho) \quad (53)$$

$$\mathfrak{M} \models \epsilon_1 \sqsubseteq \epsilon_2 \equiv \mathbb{I}(\epsilon_1) \subseteq \mathbb{I}(\epsilon_2) \quad (54)$$

$$\mathfrak{M} \models \epsilon_1 \equiv \epsilon_2 \equiv \mathbb{I}(\epsilon_1) = \mathbb{I}(\epsilon_2) \quad (55)$$

gdzie: ϵ_1, ϵ_2 reprezentują nazwy pojęć lub ról.

Spełnianie w \mathcal{ALC} (2)

$$\models \phi \equiv \forall \mathfrak{M} \mathfrak{M} \models \phi. \quad (56)$$

Spełnianie w \mathcal{ALC} (3)

$$\mathfrak{M} \models \Delta \equiv \forall \phi \in \Delta \ \mathfrak{M} \models \phi. \quad (57)$$

Spełnianie w \mathcal{ALC} (4)

δ jest spełnialne ze względu na pudełko terminologiczne $\Delta \equiv$
istnieje taki model $\mathfrak{M} = \langle \mathbb{D}, \mathbb{I} \rangle$ że, $\mathfrak{M} \models \Delta$ oraz $\mathbb{I}(\delta) \neq \emptyset$. (58)

Spełnianie w \mathcal{ALC} (5)

Fakt

$\models \delta_1 \sqsubseteq \delta_2$ wtedy i tylko wtedy, gdy dla dowolnego pudełka terminologicznego Δ , $\delta_1 \sqcap \neg \delta_1$ nie jest spełnialne ze względu na Δ .

Metoda tablic analitycznych w \mathcal{ALC} – krok po kroku

Czy $\delta_1 \sqsubseteq \delta_2$ jest prawem logiki?

- ① Utwórz pojęcie $\delta_1 \sqcap \neg\delta_2$
- ② Wstaw $(\delta_1 \sqcap \neg\delta_2)(\alpha)$ jako korzeń tablicy analitycznej T (dla dowolnej stałej α)!
- ③ Korzystając z atomowych tablic analitycznych zredukuj T !
 - ▶ **TAK**, $\delta_1 \sqsubseteq \delta_2$ **jest** prawem logiki.
 - ▶ **NIE**, $\delta_1 \sqsubseteq \delta_2$ **nie jest** prawem logiki.
- ④ Czy T jest zamknięta?

Zamknięte i otwarte tablice analityczne w logikach opisowych

Definicja

Tablica analityczna jest zamknięta, jeżeli każda jej gałąź:

- ① *zawiera dwa wyrażenia sprzeczne: $\delta(\alpha)$, $(-\delta)(\alpha)$;*
- ② *lub zawiera wyrażenie $(-\top)(\alpha)$.*

Tablica analityczna jest otwarta, jeżeli nie jest zamknięta.

Tablice analityczne w \mathcal{ALC}

- TON:

$$\begin{array}{c} -- \delta(\alpha) \\ | \\ \delta(\alpha) \end{array}$$

Tablice analityczne w \mathcal{ALC} (2)

- TOK:

$$\begin{array}{c} (\delta_1 \sqcap \delta_2)(\alpha) \\ | \\ \delta_1(\alpha) \\ | \\ \delta_2(\alpha) \end{array}$$

- TONK:

$$\begin{array}{c} -(\delta_1 \sqcap \delta_2)(\alpha) \\ \swarrow \quad \searrow \\ (-\delta_1)(\alpha) \quad (-\delta_2)(\alpha) \end{array}$$

Tablice analityczne w \mathcal{ALC} (3)

- $\text{T} \circledcirc \exists$:

$$\begin{array}{c} (\exists R.\delta)(\alpha) \\ | \\ R(\alpha, \beta) \\ | \\ \delta(\beta) \end{array}$$

gdzie: β jest nową stałą.

- $\text{T} \circledcirc \forall$:

$$\begin{array}{c} (\forall R.\delta)(\alpha) \\ | \\ R(\alpha, \beta) \\ | \\ \delta(\beta) \end{array}$$

gdzie: β jest dowolną stałą.

Tablice analityczne w \mathcal{ALC} (4)

- $\text{T}\otimes\exists:$ $(-\exists R.\delta)(\alpha)]$
 $\qquad\qquad\qquad |$
 $\qquad\qquad\qquad (\forall R. - \delta)(\alpha)$
- $\text{T}\otimes\forall:$ $(-\forall R.\delta)(\alpha)]$
 $\qquad\qquad\qquad |$
 $\qquad\qquad\qquad (\exists R. - \delta)(\alpha)$

Metoda tablic analitycznych w \mathcal{ALC} – krok po kroku (2)

Czy $\delta_1 = \delta_2$ jest prawem logiki?

- ① Utwórz pojęcia: $\delta_1 \sqcap \neg\delta_2$ oraz $\delta_2 \sqcap \neg\delta_1$
- ② Wstaw $(\delta_1 \sqcap \neg\delta_2)(\alpha)$ i $(\delta_2 \sqcap \neg\delta_1)(\alpha)$ jako korzenie dwóch tablic analitycznych T_1 i T_2 (dla dowolnej stałej α)!
- ③ Korzystając z atomowych tablic analitycznych zredukuj obie tablice!
- ④ Czy T_1 i T_2 są (zarazem) zamknięte?
 - ▶ TAK, $\delta_1 = \delta_2$ jest prawem logiki.
 - ▶ NIE, $\delta_1 = \delta_2$ nie jest prawem logiki.

Metoda tablic analitycznych w \mathcal{ALC} – krok po kroku (3)

Czy $\delta(\alpha)$ jest prawem logiki?

- ① Wstaw $(-\delta)(\alpha)$ jako korzeń tablicy analitycznej T !
- ② Korzystając z atomowych tablic analitycznych zredukuj T !
- ③ Czy T jest zamknięta?
 - ▶ TAK, $\delta(\alpha)$ jest prawem logiki.
 - ▶ NIE, $\delta(\alpha)$ nie jest prawem logiki.

Metoda tablic analitycznych w \mathcal{ALC} – krok po kroku (4)

Czy $\rho_1 \sqsubseteq \rho_2$ i $\rho_1 \equiv \rho_2$ są prawami logiki?

Czy $\rho(\alpha_1, \alpha_2)$ jest prawem logiki?

Relacje w życiu i w logice

miłość	x kocha y	Jan kocha Annę. Tomasz kocha Ewę. Tomasz kocha Annę. Tomasz kocha Jana. ...
ojcostwo	x spłodził y	Jan spłodził Zofię. ...
bycie większym	x jest większe niż y	...
bycie dłużnikiem	x jest dłużnikiem y	...
pożyczanie pieniędzy	x pożyczył od y z zł	Jan pożyczył od Tomasza 100 zł.

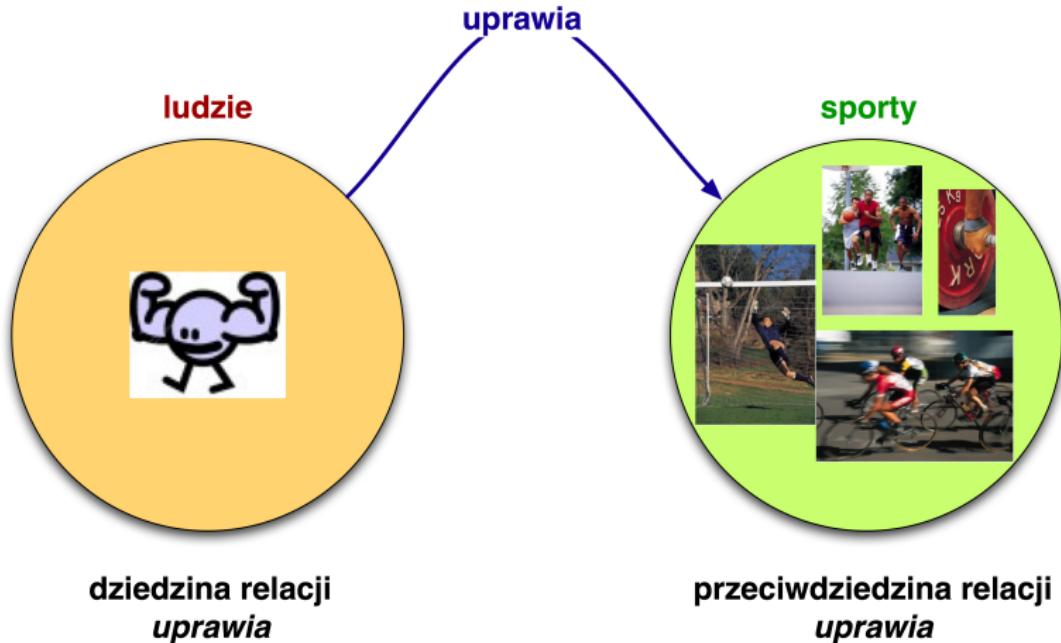
Reprezentacja relacji w WRP

- $P(x, y)$:
 - ① x kocha y
 - ② x spłodził y
 - ③ x jest większe niż y
 - ④ x jest dłużnikiem y
 - ⑤ ...
- $R(x, y, z)$:
 - ① x pożyczył od y sumę z zł
 - ② ...

Reprezentacja relacji w modelach WRP

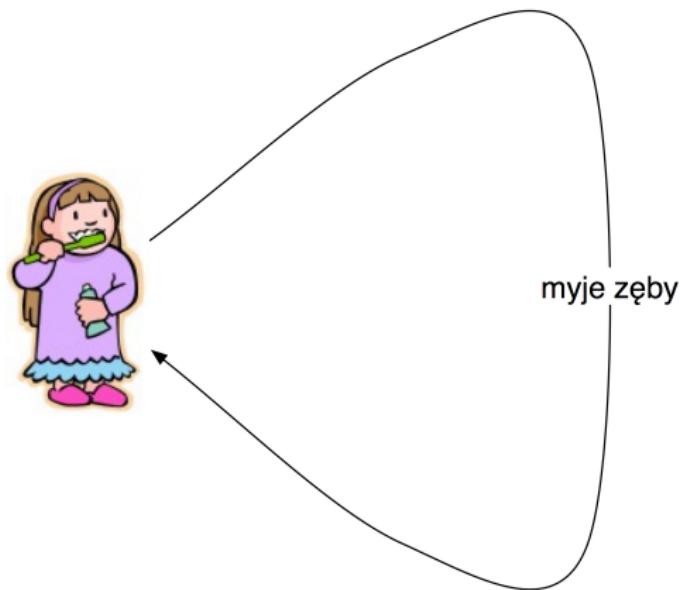
- relacje dwuargumentowe są zbiorami par (uporządkowanych)
- relacje trzyargumentowe są zbiorami trójek (uporządkowanych)
- ...
- relacje n -argumentowe są zbiorami n -tek (uporządkowanych)

Obrazkowa teoria relacji



Rysunek: Dziedzina i przeciwdziedzina relacji

Obrazkowa teoria relacji



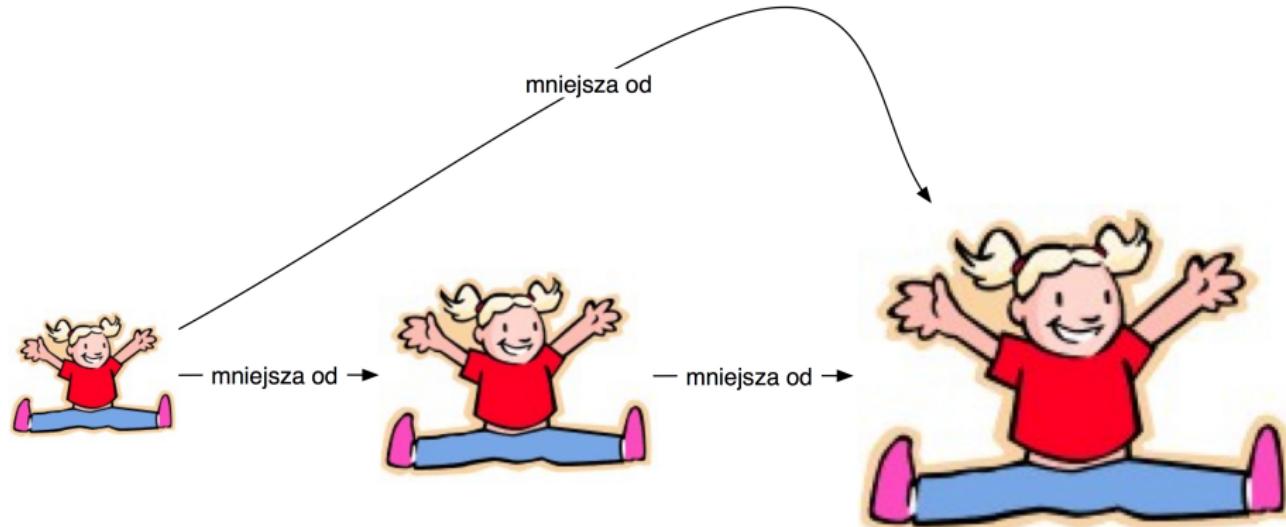
Rysunek: Zwrotność

Obrazkowa teoria relacji



Rysunek: Symetryczność

Obrazkowa teoria relacji



Rysunek: Przechodność

Dziedzina i przeciwdziedzina relacji

Definicja

Jeżeli R jest relacją dwuargumentową, to x należy do dziedziny relacji R wtedy i tylko wtedy, gdy

$$\exists y R(x, y).$$

Definicja

Jeżeli R jest relacją dwuargumentową, to x należy do przeciwdziedziny relacji R wtedy i tylko wtedy, gdy

$$\exists y R(y, x).$$

Dziedzina i przeciwdziedzina relacji - przykłady

relacja	dziedzina	przeciwdziedzina
miłość	(ludzie) kochający	(ludzie) kochani
ojcostwo	ojcowie	(prawie wszyscy) ludzie
bycie dłużnikiem	dłużnicy	wierzyciele

(Najważniejsze) rodzaje relacji dwuargumentowych

- ① zwrotne i przeciwwzrotne
- ② symetryczne, asymetryczne, antysymetryczne
- ③ przechodnie
- ④ funkcje

Relacje zwrotne

Definicja

Relacja dwuargumentowa R jest zwrotna wtedy i tylko wtedy, gdy

$$\forall x \ R(x, x).$$

przykłady:

- ① równoległość (prostych)
- ② $=$
- ③ podobieństwo

antyprzykłady:

- ① prostopadłość (prostych)
- ② $<$
- ③ bycie dłużnikiem
- ④ nienawiść

Relacje przeciwwrotne

Definicja

Relacja dwuargumentowa R jest przeciwwrotna wtedy i tylko wtedy, gdy

$$\forall x \neg R(x, x).$$

przykłady:

- ① prostopadłość (prostych)
- ② $<$
- ③ bycie dłużnikiem

antyprzykłady:

- ① równoległość (prostych)
- ② $=$
- ③ podobieństwo
- ④ nienawiść

Relacje symetryczne

Definicja

Relacja dwuargumentowa R jest symetryczna wtedy i tylko wtedy, gdy

$$\forall x, y [R(x, y) \rightarrow R(y, x)].$$

przykłady:

- ① równoległość (prostych)
- ② $=$
- ③ podobieństwo
- ④ prostopadłość (prostych)

antyprzykłady:

- ① $<$
- ② bycie dłużnikiem

Relacje asymetryczne

Definicja

Relacja dwuargumentowa R jest asymetryczna wtedy i tylko wtedy, gdy

$$\forall x, y [R(x, y) \rightarrow \neg R(y, x)].$$

przykłady:

- ① $<$
- ② bycie dłużnikiem

antyprzykłady:

- ① równoległość (prostych)
- ② prostopadłość (prostych)
- ③ $=$
- ④ podobieństwo

Relacje antysymetryczne

Definicja

Relacja dwuargumentowa R jest antysymetryczna wtedy i tylko wtedy, gdy

$$\forall x, y [R(x, y) \wedge R(y, x) \rightarrow x = y].$$

przykłady:

- ① \leq

antyprzykłady:

- ② równoległość (prostych)
- ③ prostopadłość (prostych)
- ④ podobieństwo
- ⑤ $<$
- ⑥ bycie dłużnikiem

Relacje przechodnie

Definicja

Relacja dwuargumentowa R jest przechodnia wtedy i tylko wtedy, gdy

$$\forall x, y, z [R(x, y) \wedge R(y, z) \rightarrow R(x, z)].$$

przykłady:

- ① równoległość (prostych)
- ② $<$
- ③ $=$

antyprzykłady:

- ① prostopadłość (prostych)
- ② podobieństwo

Funkcje

Definicja

Relacja dwuargumentowa R jest funkcją wtedy i tylko wtedy, gdy

$$\forall x, y_1, y_2 [R(x, y_1) \wedge R(x, y_2) \rightarrow y_1 = y_2].$$

przykłady:

- ① $=$
- ② x urodził się w dniu y
- ③ x ma numer PESEL y

antyprzykłady:

- ① równoległość (prostych)
- ② prostopadłość (prostych)
- ③ podobieństwo
- ④ $<$
- ⑤ bycie dłużnikiem

Trzyczynnikowa klasyfikacja relacji - przykłady

relacja	zwrotna	symetryczna	przechodnia
podobieństwo	tak	tak	nie
$<$	nie	asymetryczna	tak
bycie dłużnikiem	nie	nie	???

Relacje (ściśle) porządkujące

Definicja

Relacja jest ściśle porządkującą (jest porządkiem) jeżeli jest asymetryczna i przechodnia.

przykłady:

- ① $<$
- ② być głupszym niż

antyprzykłady:

- ① równoległość (prostych)
- ② \leq

Relacje równoważnościowe

Definicja

Relacja jest równoważnościowa jeżeli jest zwrotna, symetryczna i przechodnia.

przykłady

- ① równoległość (prostych)
- ② równoważenie się na wadze (x waży tyle samo, co y)

antyprzykłady:

- ① \leq
- ② bycie dłużnikiem
- ③ być niegłupszym niż

Dzięki tzw. twierdzeniu o abstrakcji relacje równoważnościowe mogą służyć do definiowania własności, np.

- ① równoległość definiuje kierunek (prostej)
- ② równoważenie się na wadze definiuje ciężar

Operacje na relacjach

- ogólne (teoriomnogościowe): suma, iloczyn, różnica, dopełnienie, itd.
- lokalne:
 - ① inwers relacji
 - ② iloczyn względny relacji

Iloczyn względny relacji

Definicja

Relacja dwuargumentowa R jest iloczynem względnym relacji S_1 i S_2 wtedy i tylko wtedy, gdy

$$\forall x, y [R(x, y) \equiv \exists z(S_1(x, z) \wedge S_2(z, y))].$$

przykłady:

- ① bycie wujkiem jest iloczynem względnym relacji bycia bratem i bycia matką
- ② bycie wnukiem jest iloczynem względnym relacji bycia dzieckiem i bycia dzieckiem
- ③ $<$ jest ilocznym względnym $<$ i $<$

Inwers relacji

Definicja

Relacja dwuargumentowa R jest inwersem relacji S wtedy i tylko wtedy, gdy

$$\forall x, y [R(x, y) \equiv S(y, x)].$$

przykłady:

- ① bycie rodzicem jest inwersem relacji bycia dzieckiem
- ② $<$ jest inwersem relacji $>$
- ③ równoległość prostych jest swoim własnym inwersem

Jaka logika opisowa jest najlepsza i dlaczego jest to *SROIQ*?

- [?]
- nazwy indywidualne
- pojęcia
 - ▶ pojęcia atomiczne (włącznie z \top)
 - ▶ operacje na pojęciach
 - ★ (uniwersalna) negacja \neg oraz iloczyn \sqcap
 - ★ suma \sqcup
 - ★ $\exists R.C$ oraz $\forall R.C$
 - ★ $\geq nR.C$ - (predykat reprezentujący) zbiór takich przedmiotów, które są powiązane relacją R z przynajmniej n przedmiotami ze zbioru C
 - ★ $\leq nR.C$ - (predykat reprezentujący) zbiór takich przedmiotów, które są powiązane relacją R z conajwyżej n przedmiotami ze zbioru C
 - ★ $\exists R.\text{Self}$ - (predykat reprezentujący) iloczyn roli R oraz relacji identyczności
 - ★ operator do tworzenia tzw. nazw nominalnych (*nominals*), np. {Elvis Presley}
- role
 - ▶ rola uniwersalna U
 - ▶ identyczność (=)
 - ▶ inwersja ról ($\bar{\cdot}$)
 - ▶ (wieloargumentowy) iloczyn względny ról (\circ)
- negacja (przy-)zdaniowa: “ \sim ”

Jaka logika opisowa jest najlepsza i dlaczego jest to *SROIQ*? (2)

- zawieranie się i równoważność pojęć i ról
- stwierdzenia (*assertions*) i negacje stwierdzeń
- formalne własności ról oraz rozłączność ról

Jak powiedzieć w \mathcal{SROIQ} tego, czego nie można powiedzieć w \mathcal{ALC} ?

- Każdy kij ma dwa końce.

$$Stick \sqsubseteq (\geq 2 \text{hasEnd}.\top) \sqcap (\leq 3 \text{hasEnd}.\top)$$

- Każdy, kto zbudował dom, posadził też drzewo.

$$\exists \text{build}.House \sqsubseteq \exists \text{plant}.Tree$$

- Przyjaciele moich przyjaciół są moimi przyjaciółmi.

$$\text{hasFriend} \circ \text{hasFriend} \sqsubseteq \text{hasFriend}$$

Czego (nadal) nie można powiedzieć w \mathcal{SROIQ} ?

- Dla każdej liczby istnieje liczba od niej większa.
- Istnieje przyczyna wszystkiego.

Alfabet \mathcal{SROIQ} – DEFINICJA

Alfabet \mathcal{SROIQ} zawiera:

- ① zbiór stałych indywidualnych $\text{CONST} := \{a_1, a_2, \dots\}$,
- ② zbiór predykatów
 - ▶ jednoargumentowych (pojęć atomicznych):

$$\text{CONCEPT} := \{\top, A_1, A_2, \dots\}$$

- ▶ dwuargumentowych (ról atomicznych):

$$\text{ROLE} := \{\text{U}, =, R_1, R_2, \dots\}$$

- ③ funktry (nazwotwórcze od argumentów nazwowych): $\neg, \sqcap, \sqcup, \circ, \top, \{\dots\}$;
- ④ “kwantyfikatory”: $\exists., \forall., \geq n., \leq n., \exists.\text{Self}$
- ⑤ funktry (zdaniotwórcze od argumentów nazwowych): $\sqsubseteq, \equiv, \text{Dis}, =, \text{Refl}, \text{Irrefl}, \text{Sym}, \text{Asym}, \text{Trans}$
- ⑥ funkтор (zdaniotwórczy od argumentu zdaniowego): \sim ;
- ⑦ nawiasy: $(,)$.

Zbiór pojęć logiki \mathcal{SROIQ} – przykłady

$Tailor \sqcap Sailor$

$Tailor \sqcup Sailor$

$\geq 2 \text{hasEnd.}\top$

$\leq 3 \text{hasEnd.}\top$

$(\geq 2 \text{hasEnd.}\top) \sqcap (\leq 3 \text{hasEnd.}\top)$

$\{John\}$

$\{John\} \sqcup \{Ann\} \sqcup \{Tom\}$

Zbiór pojęć logiki \mathcal{SROIQ}

Definicja

Zbiór pojęć logiki \mathcal{SROIQ} jest najmniejszym zbiorem napisów \mathcal{SROIQ} spełniającym poniższe warunki:

- ① Każde pojęcie atomiczne jest pojęciem \mathcal{SROIQ} .
- ② Jeżeli α jest stałą nazwową, to $\{\alpha\}$ jest pojęciem.
- ③ Jeżeli δ jest pojęciem, to $\neg\delta$ jest pojęciem.
- ④ Jeżeli δ_1, δ_2 są pojęciami, to $\delta_1 \sqcap \delta_2$ i $\delta_1 \sqcup \delta_2$ są pojęciami.
- ⑤ Jeżeli ρ jest rolą i δ jest pojęciem, to $\exists\rho.\delta$, $\forall\rho.\delta$ są pojęciami.
- ⑥ Jeżeli ρ jest prostą rolą, δ jest pojęciem i n jest liczbą naturalną, to $\geq n\rho.\delta$ i $\leq n\rho.\delta$ są pojęciami.
- ⑦ Jeżeli ρ jest prostą rolą, to $\exists\rho.\text{Self}$ jest pojęciem.

Zbiór ról logiki \mathcal{SROIQ} – przykłady

hasChild

U

hasChild⁻

hasChild \circ *hasChild*

Zbiór ról logiki \mathcal{SROIQ}

Definicja

Zbiór ról logiki \mathcal{SROIQ} jest najmniejszym zbiorem napisów \mathcal{SROIQ} spełniającym poniższe warunki:

- ① Każde rola atomiczna jest rolą \mathcal{SROIQ} .
- ② Jeżeli ρ jest rolą, to ρ^- jest rolą.
- ③ Jeżeli $\rho_1, \rho_2, \dots, \rho_n$ są rolami różnymi od roli uniwersalnej U , to $\rho_1 \circ \rho_2 \circ \dots \circ \rho_n$ jest rolą (łańcuchem ról).

Język logiki \mathcal{SROIQ}

Dowolny ciąg elementów alfabetu \mathcal{SROIQ} jest napisem logiki \mathcal{SROIQ} .

Definicja

Zbiór poprawnie zbudowanych wyrażeń logiki \mathcal{SROIQ} jest sumą trzech zbiorów:

- ① maksymalnego pudełka terminologicznego ($TBox$),
- ② maksymalnego pudełka opisowego ($ABox$),
- ③ pudełka relacyjnego ($RBox$), w którym:
 - ① hierarchia ról jest $<$ -regularna (dla pewnego $<$),
 - ② zbiór asercji ról jest prosty.

Dowolne wyrażenie należące do języka logiki \mathcal{SROIQ} nazywamy aksjomatem \mathcal{SROIQ} .

Pudełko terminologiczne \mathcal{SROIQ}

Definicja

Pudełkiem terminologicznym \mathcal{SROIQ} jest najmniejszy zbiór napisów \mathcal{SROIQ} spełniający następujący warunek:

- ① *Jeśli δ_1, δ_2 są pojęciami logiki \mathcal{SROIQ} , to $\delta_1 \sqsubseteq \delta_2$ i $\delta_1 \equiv \delta_2$ należą do pudełka terminologicznego.*

Pudełko opisowe \mathcal{SROIQ}

Definicja

Pudełkiem opisowym \mathcal{SROIQ} jest najmniejszy zbiór napisów \mathcal{SROIQ} spełniający następujące warunki:

- ① Jeżeli δ jest pojęciem logiki \mathcal{SROIQ} i α jest stałą indywidualną, to $\delta(\alpha)$ należy do pudełka opisowego.
- ② Jeżeli ρ jest rolą logiki \mathcal{SROIQ} i α_1, α_2 są stałymi indywidualnymi, to $\rho(\alpha_1, \alpha_2)$ i $\sim \rho(\alpha_1, \alpha_2)$ należą do pudełka opisowego.
- ③ Jeżeli α_1, α_2 są stałymi indywidualnymi, to $\alpha_1 = \alpha_2$ i $\sim \alpha_1 = \alpha_2$ należą do pudełka opisowego.

Pudełko relacyjne \mathcal{SROIQ}

Definicja

Pudełkiem relacyjnym \mathcal{SROIQ} jest najmniejszy zbiór napisów \mathcal{SROIQ} spełniający następujące warunki:

- ① Jeżeli ρ_1, ρ_2 są rolami logiki \mathcal{SROIQ} , to $\rho_1 \sqsubseteq \rho_2$ i $\rho_1 \equiv \rho_2$ należą do pudełka relacyjnego;
- ② Jeżeli ρ_1, ρ_2 są rolami logiki \mathcal{SROIQ} , to $Dis(\rho_1, \rho_2)$ należy do pudełka relacyjnego;
- ③ Jeżeli ρ jest rolą logiki \mathcal{SROIQ} , to $Irrefl(\rho)$, $Asym(\rho)$ należą do pudełka relacyjnego.
- ④ Jeżeli ρ jest rolą logiki \mathcal{SROIQ} , to $Refl(\rho)$, $Sym(\rho)$ i $Trans(\rho)$ należą do pudełka relacyjnego.

Zbiór wyrażeń spełniających warunek 1 nazywamy *hierarchią ról*.

Zbiór wyrażeń spełniających warunki 2 lub 4 nazywamy *zbiorem asercji ról*.

Regularny porządek ról

Definicja

Niech \mathcal{R} będzie zbiorem ról logiki $SROIQ$. Przeciwna i przekształcana relacja $<$ w zbiorze \mathcal{R} jest regularnym porządkiem, gdy dla każdych $\rho_1, \rho_2 \in \mathcal{R}$

$$\rho_1 < \rho_2 \equiv \rho_1^- < \rho_2$$

Regularne aksjomaty ról

Definicja

Niech $\rho_1 \circ \rho_2 \circ \dots \circ \rho_n$ będzie rolą logiki $SROTQ$ taką że żadna z ról $\rho_1, \rho_2, \dots, \rho_n$ nie jest rolą uniwersalną. Aksjomat subsumpcji ról $\rho_1 \circ \rho_2 \circ \dots \circ \rho_n \sqsubseteq \rho$ ($\rho \neq U$) jest $<$ -regularny jeżeli zachodzi jeden z następujących warunków:

- ① $n = 1$ i $\rho_1 = \rho^-$
- ② $n = 2$ i $\rho_1 = \rho_2 = \rho$
- ③ $\rho_i < \rho$ (dla $1 \leq i \leq n$)
- ④ $\rho_1 = \rho$ i $\rho_i < \rho$ (dla $1 < i \leq n$)
- ⑤ $\rho_n = \rho$ i $\rho_i < \rho$ (dla $1 \leq i < n$)

Regularne zbiory aksjomatów ról

Definicja

Hierarchia ról jest <-regularna jeżeli każdy aksjomat w tej hierarchii jest <-regularny.

Role proste

Niech $\phi = \rho_1 \circ \rho_2 \circ \cdots \circ \rho_n \sqsubseteq \rho$ będzie aksjomatem subsumpcji ról i niech Γ' i Γ'' będą łańcuchami ról.

Niech \mathcal{A} będzie zbiorem aksjomatów subsumpcji ról.

$\sqsubseteq_{\mathcal{A}}^*$ jest zwrotno-tranzytywnym domknięciem $\sqsubseteq_{\mathcal{A}}$.

Role proste (2)

Niech $\Gamma = \rho_1 \circ \rho_2 \circ \dots \circ \rho_n$ będzie rolą. Niech \mathcal{A} będzie zbiorem aksjomatów subsumpcji ról.

Role proste (3)

Definicja

Rola ρ jest prosta ze względu na zbiór \mathcal{A} aksjomatów subsumpcji ról, gdy $\rho_1 \circ \rho_2 \circ \dots \circ \rho_n \sqsubseteq_{\mathcal{A}^-}^ \rho$ implikuje, że $n = 1$.*

Role proste (2)

Definicja

Rola ρ jest prosta ze względu na pudełko relacyjne R , gdy

- ① nie występuje po prawej stronie aksjomatów hierarchii ról w R lub
- ② jeżeli występuje po prawej stronie jakiegoś aksjomatu hierarchii ról, to dla każdego takiego aksjomatu rola, która występuje po lewej stronie, też jest prosta lub
- ③ jest inwersem roli prostej.

Role proste (3)

Definicja

Zbiór asercji ról jest prosty gdy wszystkie aksjomaty występujące w nim o postaci $\text{Irrefl}(\rho)$, $\text{Asym}(\rho)$ lub $\text{Dis}(\rho_1, \rho_2)$, zawierają tylko role proste.

OWL 2.0 spełnieniem Twoich marzeń o reprezentacji wiedzy

- OWL – *Web Ontology Language*
- rodzina języków reprezentacji wiedzy
 - ① OWL 1.0
 - ① OWL Lite
 - ② OWL DL – logika opisowa *SHOIN*
 - ③ OWL Full
 - ② OWL 1.1
 - ③ OWL 2.0 - <http://www.w3.org/TR/owl2-overview/>
 - ① OWL EL (inferencje w czasie wielomianowym)
 - ② OWL QL (technologia bazodanowa)
 - ③ OWL RL (inferencje w czasie wielomianowym na trójkach RDF)
 - ④ OWL 2 DL – logika *SRDIQ*
- rekomendacja W3C (standardy pisania i przesyłu stron WWW)
- podstawowy (=najpopularniejszy) język ontologii inżynierijnej

I to by było na tyle ...

...

Bibliografia