Podstawy logiki

KPI

18 września 2022

Spis treści

1 Klasyczny rachunek zdań

1.1 Rzut oka wstecz

1.2 $\mathbb{KRZ}$

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

1.3 Skróty

- $\mathbb{KRZ}$
- $\mathbb{PZW}_{\mathbb{KRZ}}$
- $\mathbb{WRP}$
- $\mathbb{PZW}_{\mathbb{WRP}}$

1.4 $\mathbb{PZW}_{\mathbb{KRZ}}$ w notacji BNF

$$\phi \quad ::= \quad p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \phi \equiv \phi \tag{1}$$

1.5 $\mathbb{PZW}_{\mathbb{KRZ}}$ w notacji BNF (2)

$$\begin{aligned}
< zero_digit > &::= "0" && (2)\\
< nonzero_digit > &::= "1"|"2"|"4"|"5"|"6"|"7"|"8"|"9" && (3)\\
< digit > &::= < zero_digit > \mid < nonzero_digit > && (4)\\
< index > &::= < digit > \mid < nonzero_digit >< index > && (5)\\
< atom > &::= "p" < index > && (6)\\
< \phi > &::= < atom > \mid \\
&\quad "(""\neg" < \phi > ")"| \\
&\quad "(" < \phi > " \wedge " < \phi > ")"| \\
&\quad "(" < \phi > " \vee " < \phi > ")"| \\
&\quad "(" < \phi > " \rightarrow " < \phi > ")"| \\
&\quad "(" < \phi > " \equiv " < \phi > ")" && (7)
\end{aligned}$$

1.6 Metoda tablic analitycznych

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

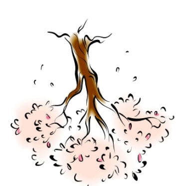
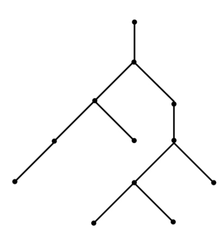
- Praw $\mathbb{KRZ}$ szukamy poprzez konstrukcję tablic analitycznych.

- Metoda tablic jest zmodyfikowaną wersją skróconej metody zero-jedynkowej.

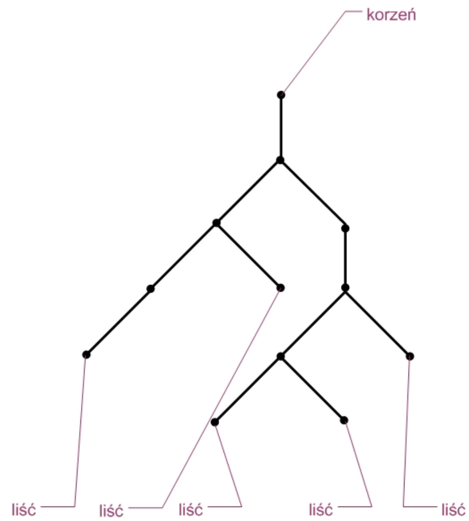
- Tablica analityczna jest binarnym drzewem znakowanym wyrażeniami $\mathbb{PZW}_{\mathbb{KRZ}}$.

1.8 Metody $\mathbb{KRZ}$ - porównanie

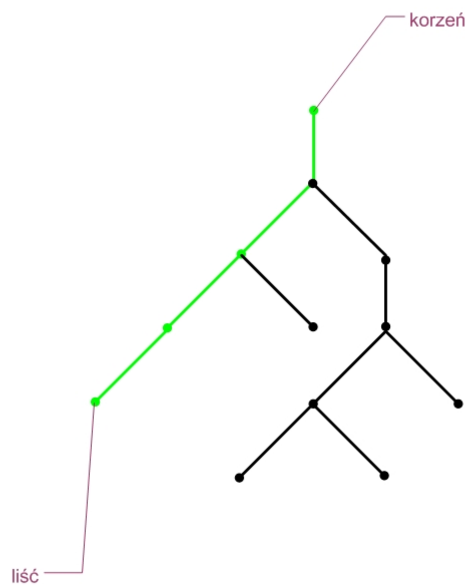
1.9 Drzewa i drzewa

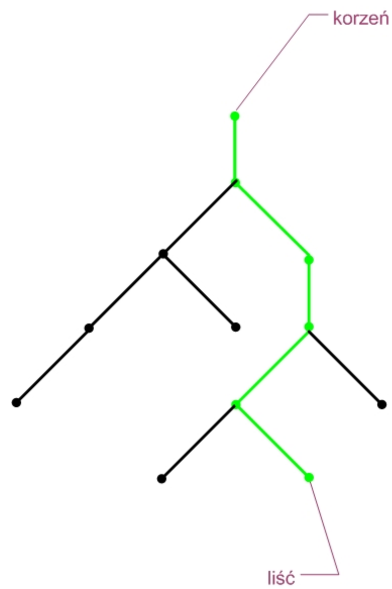
 

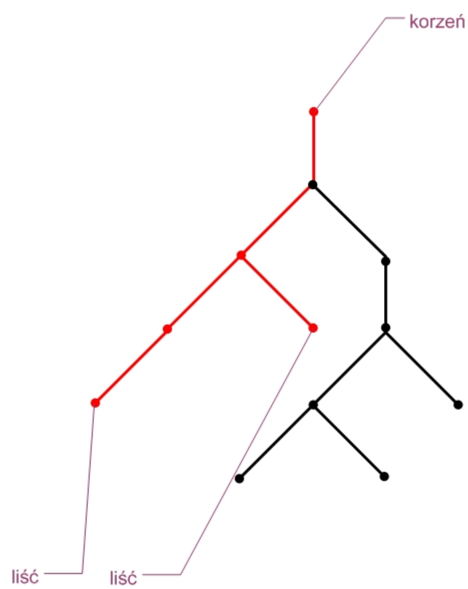
1.10 Drzewa - korzenie i liście



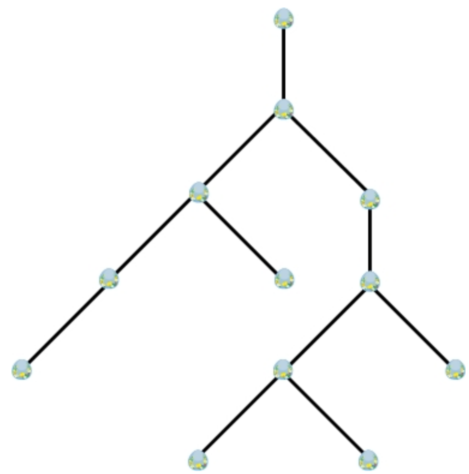
1.11 Drzewa - gałęzie, korzenie i liście

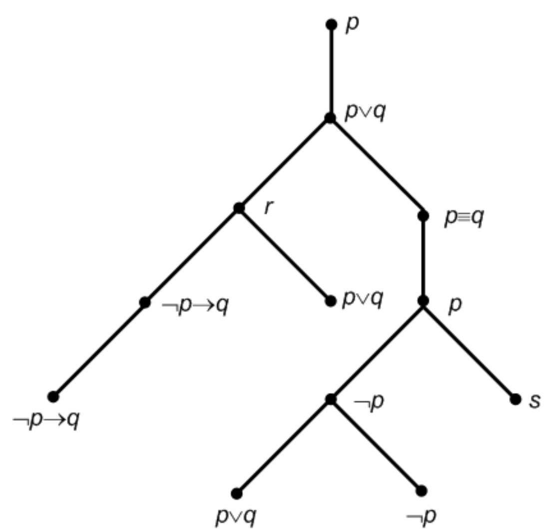



korzeń
liść

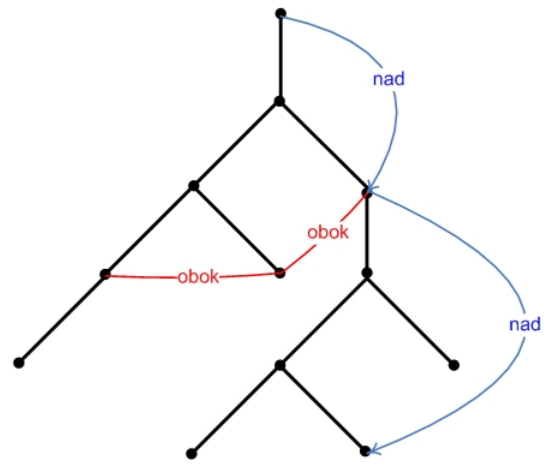

korzeń
liść
liść

1.12 Drzewa znakowane

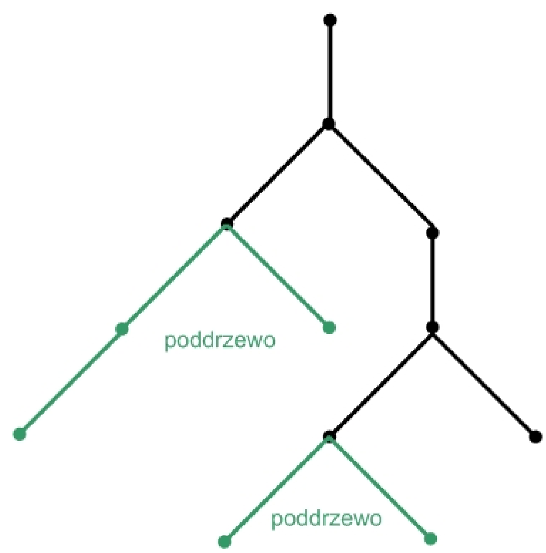


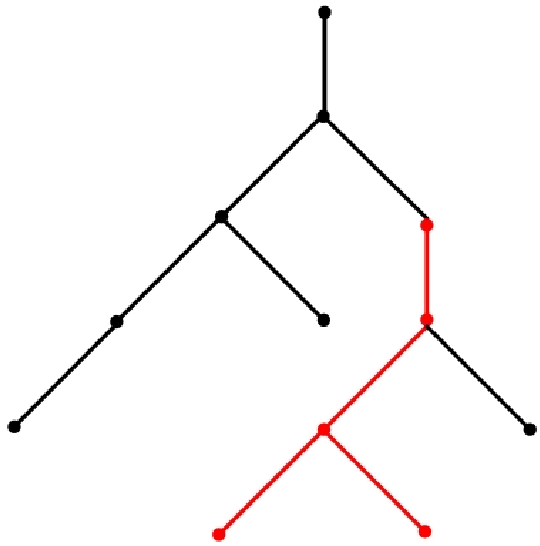


1.13 Drzewa - nad/pod i obok



1.14 Drzewa i poddrzewa





1.15 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 $\mathbb{KRZ}$ mamy jedenaście tablic atomowych:
  - $\mathbb{T}$: p
  - $\mathbb{TN}$: $\neg p$
  - $\mathbb{TON}$:
    $$\begin{array}{c} \neg\neg\phi \\ | \\ \phi \end{array}$$
- $\mathbb{TOK}$:
  $$\begin{array}{c} \phi \wedge \psi \\ | \\ \phi \\ | \\ \psi \end{array}$$
- $\mathbb{TOA}$:
  $$\begin{array}{c} \phi \vee \psi \\ \widehat{\phi \quad \psi} \end{array}$$

- $\mathbb{TOC}$:
$$\begin{array}{ccc} & \phi \to \psi & \\ \diagup & & \diagdown \\ \neg\phi & & \psi \end{array}$$

- $\mathbb{TOE}$:
$$\begin{array}{ccc} & \phi \equiv \psi & \\ \diagup & & \diagdown \\ \phi & & \neg\phi \\ | & & | \\ \psi & & \neg\psi \end{array}$$

- $\mathbb{TONK}$:
$$\begin{array}{ccc} & \neg(\phi \wedge \psi) & \\ \diagup & & \diagdown \\ \neg\phi & & \neg\psi \end{array}$$

- $\mathbb{TONA}$:
$$\begin{array}{c} \neg(\phi \vee \psi) \\ | \\ \neg\phi \\ | \\ \neg\psi \end{array}$$

- $\mathbb{TONC}$:
$$\begin{array}{c} \neg(\phi \to \psi) \\ | \\ \phi \\ | \\ \neg\psi \end{array}$$

- $\mathbb{TONE}$:
$$\begin{array}{ccc} & \neg(\phi \equiv \psi) & \\ \diagup & & \diagdown \\ \phi & & \neg\phi \\ | & & | \\ \neg\psi & & \psi \end{array}$$

1.16 Zredukowane tablice analityczne

- Tablice analityczne konstruujemy tak długo, aż uzyskamy zredukowaną ich postać.

- Jeżeli ϕ jest wyrażeniem $\mathbb{KRZ}$, tj. gdy należy do $\mathbb{PZW}_{\mathbb{KRZ}}$, to symbol T_ϕ oznacza drzewo atomowe, które ma w korzeniu ϕ.

Definicja 1 *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.*

1.17 Zamknięte i otwarte tablice analityczne

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

Definicja 2 *Tablice analityczna jest* zamknięta*, jeżeli każda jej gałąź zawiera dwa wyrażenia sprzeczne: ϕ, $\neg\phi$.*
Tablice analityczna jest otwarta*, jeżeli nie jest zamknięta.*

Fakt 1 *Wyrażenie $\phi \in \mathbb{PZW}_{\mathbb{KRZ}}$ jest prawem logiki wtedy i tylko wtedy, gdy istnieje zamknięta tablica analityczna dla $\neg\phi$.*

1.18 Metoda tablic analitycznych - krok po kroku

Czy $\phi \in \mathbb{PZW}_{\mathbb{KRZ}}$ 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 $\Longrightarrow$ ϕ jest prawem logiki.
   - NIE $\Longrightarrow$ ϕ nie jest prawem logiki.

1.19 Metoda tablic analitycznych - przykłady

¡1¿ Czy wyrażenie $(p \to q) \to p \vee q$ jest prawem logiki?
¡2¿ $\neg((p \to q) \to p \vee q)$ ¡3¿ $\begin{array}{c} \neg((p \to q) \to p \vee q) \\ | \\ p \to q \\ | \\ \neg(p \vee q) \end{array}$ ¡4¿ $\begin{array}{c} \neg((p \to q) \to p \vee q) \\ | \\ p \to q \\ | \\ \neg(p \vee q) \\ | \\ \neg p \\ | \\ \neg q \end{array}$

¡5¿

$$\neg((p \to q) \to p \vee q)$$
$$|$$
$$p \to q$$
$$|$$
$$\neg(p \vee q)$$
$$|$$
$$\neg p$$
$$|$$
$$\neg q$$
$$\begin{array}{cc} \diagup & \diagdown \\ \neg p & q \end{array}$$

1.20 Metoda tablic analitycznych - przykłady (2)

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

$$\neg((p \to q) \to p \vee q)$$
$$|$$
$$p \to q$$
$$|$$
$$\neg(p \vee q)$$
$$|$$
$$\neg p$$
$$|$$
$$\neg q$$
$$\diagup$$
$$\neg p$$

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

1.21 Metoda tablic analitycznych - przykłady (3)

¡1¿ Czy wyrażenie $q \to (p \to q)$ jest prawem logiki?

¡2¿ $\neg(q \to (p \to q))$

¡3¿

$$\neg(q \to (p \to q))$$
$$|$$
$$q$$
$$|$$
$$\neg(p \to q)$$

¡4¿

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

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

1.22 Modele i kontrmodele

- model i kontrmodel formuły
- modele a drzewa

1.23 Automatyzacja metody drzew/tablic

- `https://www.umsu.de/trees/`

1.24 Metalogiczne własności $\mathbb{KRZ}$

1.25 Metalogiczne własności $\mathbb{KRZ}$

1. $\mathbb{KRZ}$ jest niesprzeczny
2. $\mathbb{KRZ}$ jest (zu)pełny
3. $\mathbb{KRZ}$ jest rozstrzygalny

1.26 Tezy i tautologie w logice

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

1.27 Niesprzeczność 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 $\neg\phi$.

1.28 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.

1.29 Rozstrzygalność teorii logicznej

- Teoria logiczna jest *rozstrzygalna*, gdy istnieje metoda, która w skończonej liczbie ściśle 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.

2 Węższy rachunek predykatów

2.1 Motywacja

2.2 Poza $\mathbb{KRZ}$

Pewne wnioskowania dedukcyjne są poza zasięgiem $\mathbb{KRZ}$:

- wnioskowania łatwe, np.:

$$\frac{\text{Żaden gryzoń nie jest gadem.}}{\text{Nieprawda, że niektóre gady są gryzoniami.}}$$

- wnioskowania trudne, np.:

$$\frac{\begin{array}{c}\text{Dłużnicy naszych dłużników są naszymi dłużnikami.}\\ \text{Każdy jest dłużnikiem swego każdego wierzyciela.}\\ \text{Każdy jest wierzycielem swego każdego dłużnika.}\end{array}}{\text{Wierzyciele naszych wierzycieli są naszymi wierzycielami.}}$$

2.3 Porównanie języków

2.4 Język $\mathbb{WRP}$

2.5 Predykaty

predykaty:

- jednoargumentowe:
  - …umiera,
  - …jest nieparzysta.
- wieloargumentowe:
  - $\dots < \dots$,
  - $\dots \in \dots$,
  - …kocha …,
  - …leży pomiędzy …a ….

2.6 Alfabet $\mathbb{WRP}$

Definicja 3 *Alfabet $\mathbb{WRP}$ 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:* $P, Q, R, \ldots, P_1, \ldots,$

4. *zbiór funktorów prawdziwościowych:* $\neg, \wedge, \vee, \rightarrow, \equiv,$

5. *zbiór kwantyfikatorów:* $\forall, \exists,$

6. *nawiasy: (,).*

2.7 Język $\mathbb{WRP}$

Sumę zbiorów:

1. zmiennych indywidualnych

2. stałych indywidualnych

będziemy nazywać zbiorem *wyrażeń nazwowych* $\mathbb{WRP}$.
Dowolny ciąg elementów alfabetu $\mathbb{WRP}$ jest *napisem* ($\mathbb{WRP}$).

2.8 Język $\mathbb{WRP}(2)$

Definicja 4 *Zbiór poprawnie zbudowanych wyrażeń $\mathbb{WRP}$ (skrót: $\mathbb{PZW}_{\mathbb{WRP}}$) jest najmniejszym zbiorem napisów spełniającym poniższe warunki:*

1. *Jeżeli $\delta^{(n)}$ jest predykatem n-argumentowym ($n \geq 1$), a $\alpha_1, \alpha_2, \ldots, \alpha_n$ są wyrażeniami nazwowymi, to $\delta^{(n)}(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{PZW}_{\mathbb{WRP}}$.*

2. *Jeśli $\phi \in \mathbb{PZW}_{\mathbb{WRP}}$, to $\neg\phi \in \mathbb{PZW}_{\mathbb{WRP}}$.*

3. *Jeśli $\phi, \psi \in \mathbb{PZW}_{\mathbb{WRP}}$, to $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi), (\phi \equiv \psi) \in \mathbb{PZW}_{\mathbb{WRP}}$.*

4. *Jeżeli $\phi \in \mathbb{PZW}_{\mathbb{WRP}}$ i α jest zmienną indywidualną, to $\forall\alpha(\phi)$ i $\exists\alpha(\phi) \in \mathbb{PZW}_{\mathbb{WRP}}$.*

2.9 Alfabet $\mathbb{WRP}$ – notacja BNF

$$< zero\_digit > ::= "0" \tag{8}$$

$$< nonzero\_digit > ::= "1" | "2" | "4" | "5" | "6" | "7" | "8" | "9" \tag{9}$$

$$< digit > ::= < zero\_digit > \mid < nonzero\_digit > \tag{10}$$

$$< index > ::= < digit > \mid < nonzero\_digit >< index > \tag{11}$$

$$< const > ::= "a" < index > \tag{12}$$

$$< var > ::= "x" < index > \tag{13}$$

$$< \alpha > ::= < const > \mid < var > \tag{14}$$

$$< seq > ::= < \alpha > \mid < \alpha > "," < seq > \tag{15}$$

$$< pred > ::= "P" < index > \tag{16}$$

2.10 Język $\mathbb{WRP}$ – notacja BNF

$$< atom > ::= < pred > "(" < seq > ")" \tag{17}$$

$$\begin{aligned} < \phi > ::= & < atom > \mid \\ & "\forall" < var > \phi \mid \\ & "\exists" < var > \phi \mid \\ & "\neg" < \phi > \mid \\ & < \phi > " \wedge " < \phi > \mid \\ & < \phi > " \vee " < \phi > \mid \\ & < \phi > " \rightarrow " < \phi > \mid \\ & < \phi > " \equiv " < \phi > \end{aligned} \tag{18}$$

2.11 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.

2.12 Notacja nawiasowa zredukowana w $\mathbb{WRP}$

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

2.13 Język $\mathbb{WRP}$ - przykłady

2.14 Metoda tablic analitycznych

2.15 Metoda tablic analitycznych w $\mathbb{WRP}$

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

2.16 Ku nowym regułom - krok po kroku

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

2.17 Zasięg kwantyfikatorów

Definicja 5 *W wyrażeniach* $\forall\alpha\ \phi$ *i* $\exists\alpha\ \phi$ *wyrażenie* ϕ *jest* zasięgiem *kwantyfikatora: ogólnego lub szczegółowego.*

2.18 Wolność i niewola zmiennych

Definicja 6 *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 7 *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*.*

2.19 Wolność i niewola zmiennych - przykłady

2.20 Operacja podstawiania – ogólnie

Definicja 8 *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:

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

Wiedza	$\mathbb{KRZ}$	**sylogistyka**	$\mathbb{WRP}$
Jan jest bogaty	p	-	$A(\mathrm{a})$
Warszawa jest stolicą Polski	p	-	$P(\mathrm{a}, \mathrm{b})$
Jan jest bogaty lub uczciwy	$p \vee q$	-	$A(\mathrm{a}) \vee B(\mathrm{a})$
Anna lub Jan są bogaci	$p \vee q$	-	$A(\mathrm{a}) \vee A(\mathrm{b})$
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	-	$\forall x\ (P(x, \mathrm{a}) \rightarrow Q(x, \mathrm{b}))$

Tabela 1: Wiedza w językach logiki

Tabela 2: Przykłady poprawnie i niepoprawnie zbudowanych $\mathbb{PZW}_{\mathbb{WRP}}$

$\mathbb{PZW}_{\mathbb{WRP}}$	**non-**$\mathbb{PZW}_{\mathbb{WRP}}$
$A(x)$	$A(x, x)$
$P(\mathrm{a}, \mathrm{b})$	$P()$
$\forall x\ A(x)$	$A(\forall x)$
$\forall x \forall x\ A(x)$	
$\forall x \exists y\ P(z_1, z_2)$	$\forall x \exists y\ P(z_1,)$

Tabela 3: Przykłady wolnych i związanych zmiennych w $\mathbb{PZW}_{\mathbb{WRP}}$

$\mathbb{PZW}_{\mathbb{KRZ}}$	**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

2.21 Operacja podstawiania

Definicja 9 *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:

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

2.22 Operacja podstawiania - przykłady

2.23 Nowe atomowe tablice analityczne w $\mathbb{WRP}$

- $\mathbb{TO}\forall$:
$$\begin{array}{c} \forall\alpha\ \phi \\ | \\ \phi(\alpha/\xi) \end{array}$$
gdzie: ξ jest dowolną stałą.

- $\mathbb{TO}\exists$:
$$\begin{array}{c} \exists\alpha\ \phi \\ | \\ \phi(\alpha/\xi) \end{array}$$
gdzie: ξ jest dowolną **NOWĄ** stałą, tj. stałą niewystępującą w danej gałęzi.

2.24 Nowe atomowe tablice analityczne w $\mathbb{WRP}$ (2)

- $\mathbb{TON}\forall$:
$$\begin{array}{c} \neg\forall\alpha\ \phi \\ | \\ \neg\phi(\alpha/\xi) \end{array}$$
gdzie: ξ jest dowolną **NOWĄ** stałą, tj. stałą niewystępującą w danej gałęzi.

- $\mathbb{TON}\exists$:
$$\begin{array}{c} \neg\exists\alpha\ \phi \\ | \\ \neg\phi(\alpha/\xi) \end{array}$$
gdzie: ξ jest dowolną stałą.

2.25 Metoda tablic analitycznych - przykłady

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

¡2¿ $\neg(\forall x\ A(x) \rightarrow \exists x\ A(x))$

¡3¿

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

¡4¿

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

¡5¿

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

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

2.26 Metoda tablic analitycznych - przykłady (2)

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

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

¡3¿

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

¡4¿

$$\neg(\exists x \forall y\ P(x,y) \to \forall y \exists x\ P(x,y))$$
$$|$$
$$\exists \mathbf{x} \forall \mathbf{y}\ \mathbf{P}(\mathbf{x},\mathbf{y})$$
$$|$$
$$\neg\forall y \exists x\ P(x,y)$$
$$|$$
$$\forall \mathbf{y}\ \mathbf{P}(\mathbf{a},\mathbf{y})$$

¡5¿

$$\neg(\exists x \forall y\ P(x,y) \to \forall y \exists x\ P(x,y))$$
$$|$$
$$\exists x \forall y\ P(x,y)$$
$$|$$
$$\neg\forall \mathbf{y} \exists \mathbf{x}\ \mathbf{P}(\mathbf{x},\mathbf{y})$$
$$|$$
$$\forall y\ P(a,y)$$
$$|$$
$$\neg\exists \mathbf{x}\ \mathbf{P}(\mathbf{x},\mathbf{b})$$

¡6¿

$$\neg(\exists x \forall y\ P(x,y) \to \forall y \exists x\ P(x,y))$$
$$|$$
$$\exists x \forall y\ P(x,y)$$
$$|$$
$$\neg\forall y \exists x\ P(x,y)$$
$$|$$
$$\forall \mathbf{y}\ \mathbf{P}(\mathbf{a},\mathbf{y})$$
$$|$$
$$\neg\exists x\ P(x,b)$$
$$|$$
$$\mathbf{P}(\mathbf{a},\mathbf{b})$$

¡7¿

$$\neg(\exists x \forall y\ P(x,y) \to \forall y \exists x\ P(x,y))$$
$$|$$
$$\exists x \forall y\ P(x,y)$$
$$|$$
$$\neg\forall y \exists x\ P(x,y)$$
$$|$$
$$\forall y\ P(a,y)$$
$$|$$
$$\neg\exists \mathbf{x}\ \mathbf{P}(\mathbf{x},\mathbf{b})$$
$$|$$
$$P(a,b)$$
$$|$$
$$\neg\mathbf{P}(\mathbf{a},\mathbf{b})$$

¡8¿

$$\neg(\exists x \forall y\ P(x,y) \to \forall y \exists x\ P(x,y))$$
|
$$\exists x \forall y\ P(x,y)$$
|
$$\neg \forall y \exists x\ P(x,y)$$
|
$$\forall y\ P(a,y)$$
|
$$\neg \exists x\ P(x,b)$$
|
$$P(a,b)$$
|
$$\neg P(a,b)$$
|
$$P(a,a)$$
|
$$P(b,b)$$

¡9¿ Tak, wyrażenie $\exists x \forall y\ P(x,y) \to \forall y \exists x\ P(x,y)$ jest prawem logiki.

2.27 Metoda tablic analitycznych - przykłady (3)

¡1¿ Czy wyrażenie $\forall x\ (A(x) \vee B(x)) \to \forall x\ A(x) \vee \forall x\ B(x)$ jest prawem logiki?

¡2¿

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

¡3¿

$$\neg(\forall x\ (A(x) \vee B(x)) \to \forall x\ A(x) \vee \forall x\ B(x))$$
|
$$\forall x\ (A(x) \vee B(x))$$
|
$$\neg(\forall x\ A(x) \vee \forall x\ A(x))$$

¡4¿

$$\neg(\forall x\ (A(x) \vee B(x)) \to \forall x\ A(x) \vee \forall x\ B(x))$$
|
$$\forall x\ (A(x) \vee B(x))$$
|
$$\neg(\forall x\ A(x) \vee \forall x\ A(x))$$
|
$$\neg \forall x\ A(x)$$
|
$$\neg \forall x\ B(x)$$

¡5¿

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

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

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

$$\neg\forall \mathbf{x}\ \mathbf{A}(\mathbf{x})$$

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

$$\neg\mathbf{A}(\mathbf{a})$$

¡6¿

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

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

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

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

$$\neg\forall \mathbf{x}\ \mathbf{B}(\mathbf{x})$$

$$\neg A(a)$$

$$\neg\mathbf{B}(\mathbf{b})$$

¡7¿

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

$$\forall \mathbf{x}\ (\mathbf{A}(\mathbf{x}) \vee \mathbf{B}(\mathbf{x}))$$

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

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

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

$$\neg A(a)$$

$$\neg B(b)$$

$$\mathbf{A}(\mathbf{a}) \vee \mathbf{B}(\mathbf{a})$$

$$\mathbf{A}(\mathbf{b}) \vee \mathbf{B}(\mathbf{b})$$

¡8¿

$$\forall x\ (A(x) \lor B(x))$$

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

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

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

$$\neg A(a)$$

$$\neg B(b)$$

$$A(a) \lor B(a)$$

$$A(b) \lor B(b)$$

$$A(a) \qquad B(a)$$

¡9¿

$$\neg(\forall x\ A(x) \lor \forall x\ A(x))$$

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

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

$$\neg A(a)$$

$$\neg B(b)$$

$$A(a) \lor B(a)$$

$$A(b) \lor B(b)$$

$$A(a) \qquad\qquad B(a)$$

$$A(b) \quad B(b) \qquad A(b) \quad B(b)$$

¡10¿

$$\neg(\forall x\ A(x) \vee \forall x\ A(x))$$
$$|$$
$$\neg\forall x\ A(x)$$
$$|$$
$$\neg\forall x\ B(x)$$
$$|$$
$$\neg A(a)$$
$$|$$
$$\neg B(b)$$
$$|$$
$$A(a) \vee B(a)$$
$$|$$
$$A(b) \vee B(b)$$
$$\begin{array}{cc} / & \backslash \\ ... & B(a) \end{array}$$
$$\begin{array}{cccc} / & \backslash & / & \backslash \\ ... & ... & A(b) & ... \end{array}$$

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

2.28 Metoda tablic analitycznych - przykłady (4)

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

¡2¿

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

¡3¿

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

¡4¿

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

¡5¿

$$\neg(\forall y \exists x\ P(x,y) \rightarrow \exists x \forall y\ P(x,y)$$
$$\forall y \exists x\ P(x,y)$$
$$\neg\exists x \forall y\ P(x,y)$$
$$\exists \mathbf{x}\ \mathbf{P}(\mathbf{x},\mathbf{a})$$
$$\mathbf{P}(\mathbf{b},\mathbf{a})$$

¡6¿

$$\neg(\forall y \exists x\ P(x,y) \rightarrow \exists x \forall y\ P(x,y)$$
$$\forall y \exists x\ P(x,y)$$
$$\neg\exists \mathbf{x} \forall \mathbf{y}\ \mathbf{P}(\mathbf{x},\mathbf{y})$$
$$\exists x\ P(x,a)$$
$$P(b,a)$$
$$\neg\forall \mathbf{y}\ \mathbf{P}(\mathbf{a},\mathbf{y})$$

¡7¿

$$\neg(\forall y \exists x\ P(x,y) \rightarrow \exists x \forall y\ P(x,y)$$
$$\forall y \exists x\ P(x,y)$$
$$\neg\exists \mathbf{x} \forall \mathbf{y}\ \mathbf{P}(\mathbf{x},\mathbf{y})$$
$$\exists x\ P(x,a)$$
$$P(b,a)$$
$$\neg\forall y\ P(a,y)$$
$$\neg\forall \mathbf{y}\ \mathbf{P}(\mathbf{b},\mathbf{y})$$

¡8¿

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

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

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

$$\exists x\ P(x,a)$$

$$P(b,a)$$

$$\neg \forall \mathbf{y}\ \mathbf{P}(\mathbf{a},\mathbf{y})$$

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

$$\neg \mathbf{P}(\mathbf{a},\mathbf{c})$$

¡9¿

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

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

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

$$\exists x\ P(x,a)$$

$$P(b,a)$$

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

$$\neg \forall \mathbf{y}\ \mathbf{P}(\mathbf{b},\mathbf{y})$$

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

$$\neg \mathbf{P}(\mathbf{b},\mathbf{d})$$

¡10¿

$$\neg(\forall y \exists x\ P(x,y) \to \exists x \forall y\ P(x,y)$$
$$|$$
$$\forall \mathbf{y} \exists \mathbf{x}\ \mathbf{P}(\mathbf{x},\mathbf{y})$$
$$|$$
$$\neg\exists x \forall y\ P(x,y)$$
$$|$$
$$\exists x\ P(x,a)$$
$$|$$
$$P(b,a)$$
$$|$$
$$\neg\forall y\ P(a,y)$$
$$|$$
$$\neg\forall y\ P(b,y)$$
$$|$$
$$\neg P(a,c)$$
$$|$$
$$\neg P(b,d)$$
$$|$$
$$\exists \mathbf{x}\ \mathbf{P}(\mathbf{x},\mathbf{d})$$

¡11¿

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

- ¡12-14¿ Metoda tablic analitycznych nie jest w stanie rozstrzygnąć, czy $\forall y \exists x\ P(x,y) \to \exists x \forall y\ P(x,y)$ jest prawem logiki.

- ¡13-14¿W $\mathbb{WRP}$ metoda tablic analitycznych jest metodą półalgorytmiczną.
- ¡14¿ Nie istnieje algorytmiczna metoda szukania praw $\mathbb{WRP}$ - $\mathbb{WRP}$ jest teorią nierozstrzygalną.

- ¡15-16¿ Wyrażenie $\forall y \exists x \; P(x, y) \rightarrow \exists x \forall y \; P(x, y)$ nie jest prawem logiki.
  - ¡16¿ $P(x, y) - x > y$.

2.29 Ograniczenia metody tablic analitycznych dla $\mathbb{WRP}$

- półalgorytmiczność:
  - niektóre tablice analityczne dla nie-praw $\mathbb{WRP}$ są nieskończone
- czasami trzeba "oszukiwać":
  - tablica dla $A(x) \rightarrow \exists x \; A(x)$ nie jest zamknięta, mimo że jest to prawo $\mathbb{WRP}$,
  - 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.

2.30 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.

2.31 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

$$\frac{\forall x(A(x) \to \neg B(x))}{\neg \exists x(B(x) \land A(x))}$$

$$\forall x(A(x) \to \neg B(x)) \to \neg \exists x(B(x) \land A(x))$$

2.32 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

$$\frac{\begin{array}{c}\forall x \forall y \forall z[P(y, z) \land P(x, y) \to P(x, z)] \\ \forall x \forall y[Q(x, y) \to P(y, x)] \\ \forall x \forall y[P(x, y) \to Q(y, x)]\end{array}}{\forall x \forall y \forall z[Q(y, z) \land Q(x, y) \to Q(x, z)]}$$

2.33 Semantyka dla $\mathbb{WRP}$

2.34 Modele $\mathbb{WRP}$

Definicja 10 *Parę $< \mathbb{D}, \mathbb{I} >$ nazywamy modelem dla języka $\mathbb{WRP}$ jeżeli:*

1. $\mathbb{D} \neq \emptyset$,
2. $\mathbb{I}$ *jest taką funkcją, która:*
   (a) *odwzorowuje zbiór $\mathbb{CONST}$ stałych języka $\mathbb{WRP}$ w zbiór $\mathbb{D}$,*
   (b) *odwzorowuje zbiór predykatów **jedno**argumentowych $\mathbb{REL}$ w $\wp(\mathbb{D})$ (=zbiór wszystkich podzbiorów $\mathbb{D}$),*
   (c) *odwzorowuje zbiór predykatów **dwu**argumentowych $\mathbb{REL}$ w $\wp(\mathbb{D} \times \mathbb{D})$ (=zbiór wszystkich podzbiorów iloczynu kartezjańskiego $\mathbb{D}$ "razy" $\mathbb{D}$)*
   (d) $\ldots$,
   (e) *odwzorowuje zbiór predykatów n-argumentowych $\mathbb{REL}$ w $\wp(\mathbb{D}^n)$ (=zbiór wszystkich podzbiorów n-argumentowego iloczynu kartezjańskiego $\underbrace{\mathbb{D} \times \mathbb{D} \times \cdots \times \mathbb{D}}_{n \text{ razy}}$).*

Zbiór $\mathbb{D}$ nazywamy dziedziną modelu.

2.35 Wartościowanie w $\mathbb{WRP}$

Definicja 11 *Niech $\mathfrak{M} =< \mathbb{D}, \mathbb{I} >$ będzie modelem $\mathbb{WRP}$.*

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

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

2.36 Spełnianie w $\mathbb{WRP}$

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

- $\mathfrak{M}, \mathfrak{v} \models \phi$, to tyle, co ϕ jest spełnione w modelu $\mathfrak{M}$ przy wartościowaniu $\mathfrak{v}$.

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

- $\mathfrak{M} \models \phi$, to tyle, co ϕ jest spełnione w modelu $\mathfrak{M}$.

- $\models \phi$, to tyle, co ϕ jest tautologią.

2.37 Spełnianie w $\mathbb{WRP}$ (2)

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

$$\mathfrak{M}, \mathfrak{v} \models \neg\phi \equiv \mathfrak{M} \not\models \phi \tag{21}$$

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

$$\text{itd.}$$

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

$$\mathfrak{M}, \mathfrak{v} \models \exists\alpha\ \phi \equiv \text{ dla jakiego } x \in \mathbb{D}\ \mathfrak{M}, \mathfrak{v}[\alpha/x] \models \phi \tag{24}$$

2.38 Spełnianie w $\mathbb{WRP}$ (3)

$$\mathfrak{M} \models \phi \equiv \forall\mathfrak{v}\ \mathfrak{M}, \mathfrak{v} \models \phi. \tag{25}$$

$$\models \phi \equiv \forall\mathfrak{M}\ \mathfrak{M} \models \phi. \tag{26}$$

2.39 Spełnianie w $\mathbb{WRP}$ 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.

2.40 Semantyka dla języka PRZYKŁAD

2.41 PRZYKŁAD – Alfabet

$$< const > ::= \text{"}Donald\text{"} | \text{"Jarosław"} \tag{27}$$

$$< var > ::= \text{"}x\text{"} | \text{"}y\text{"} | \text{"}z\text{"} \tag{28}$$

$$< nominal > ::= < const > \mid < var > \tag{29}$$

$$< seq2 > ::= < nominal > \text{","} < nominal > \tag{30}$$

$$< pred > ::= \text{"jest przyjacielem"} \tag{31}$$

2.42 PRZYKŁAD – Język

$$< atom > ::= < pred > \text{"("} < seq2 > \text{")"} \tag{32}$$

$$\begin{aligned} < \phi > ::= & < atom > \mid \\ & \text{"}\forall\text{"} < var > \phi \mid \\ & \text{"}\exists\text{"} < var > \phi \mid \\ & \text{"}\neg\text{"} < \phi > \mid \\ & < \phi > \text{"} \wedge \text{"} < \phi > \mid \\ & < \phi > \text{"} \vee \text{"} < \phi > \mid \\ & < \phi > \text{"} \rightarrow \text{"} < \phi > \mid \\ & < \phi > \text{"} \equiv \text{"} < \phi > \end{aligned} \tag{33}$$

2.43 PRZYKŁAD – Model $\mathfrak{M}_1$

1. $\mathbb{D} = \{☺, ☹, ☻\}$
2. $\mathbb{D} \times \mathbb{D}$
3. $\mathbb{I}(\text{"Donald"}) = ☺$
4. $\mathbb{I}(\text{"Jarosław"}) = ☹$
5. $\mathbb{I}(\text{"jest przyjacielem"}) = \{< ☺, ☺ >, < ☺, ☹ >\}$

2.44 PRZYKŁAD – Wartościowanie $\mathfrak{v}_1$

- $\mathfrak{v}_1(\text{"x"}) = \smiley$
- $\mathfrak{v}_1(\text{"y"}) = \smiley$
- $\mathfrak{v}_1(\text{"z"}) = \smiley$

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

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

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

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

2.47 PRZYKŁAD – Wartościowanie $\mathfrak{v}_2$

- $\mathfrak{v}_1(\text{"x"}) = \smiley$
- $\mathfrak{v}_1(\text{"y"}) = \frownie$
- $\mathfrak{v}_1(\text{"z"}) = \blacksmiley$

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

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

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

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

2.50 PRZYKŁAD – Model $\mathfrak{M}_2$

1. $\mathbb{D} = \{☺, ☹, ☻\}$
2. $\mathbb{I}$("Donald") = ☺
3. $\mathbb{I}$("Jarosław") = ☹
4. $\mathbb{I}$("jest przyjacielem")$=\mathbb{D} \times \mathbb{D}$

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

- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Donald jest przyjacielem Jarosław"
- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Donald jest przyjacielem Donald"
- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Jarosław jest przyjacielem Donald"
- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Jarosław jest przyjacielem Jarosław"

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

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

2.53 PRZYKŁAD – Model $\mathfrak{M}_3$

1. $\mathbb{D} = \{$☺, ☹, ☻$\}$
2. $\mathbb{I}$("Donald") = ☺
3. $\mathbb{I}$("Jarosław") = ☹
4. $\mathbb{I}$("jest przyjacielem")=$\emptyset$

2.54 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"

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

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

2.56 Modele

2.57 PRZYKŁAD – Alfabet

$$< const >::= "Donald" | "\text{Jarosław}" \tag{34}$$
$$< var >::= "x" | "y" | "z" \tag{35}$$
$$< nominal >::=< const > | < var > \tag{36}$$
$$< seq2 >::=< nominal > "," < nominal > \tag{37}$$
$$< pred >::= "\text{jest przyjacielem}" \tag{38}$$

2.58 PRZYKŁAD – Język

$$< atom >::=< pred > "(" < seq2 > ")" \tag{39}$$
$$\begin{aligned} < \phi >::=&< atom > | \\ &"\forall" < var > \phi| \\ &"\exists" < var > \phi| \\ &"\neg" < \phi > | \\ &< \phi > " \wedge " < \phi > | \\ &< \phi > " \vee " < \phi > | \\ &< \phi > " \rightarrow " < \phi > | \\ &< \phi > " \equiv " < \phi > \end{aligned} \tag{40}$$

2.59 PRZYKŁAD – Model $\mathfrak{M}_1$

1. $\mathbb{D} = \{☺, ☹, ☻\}$
2. $\mathbb{I}("\text{Donald}") = ☺$
3. $\mathbb{I}("\text{Jarosław}") = ☹$
4. $\mathbb{I}("\text{jest przyjacielem}") = \{< ☺, ☺ >, < ☺, ☹ >\}$

2.60 PRZYKŁAD – Wartościowanie $\mathfrak{v}_1$

- $\mathfrak{v}_1("x") = ☺$

- $\mathfrak{v}_1(\text{”y”}) = ☺$
- $\mathfrak{v}_1(\text{”z”}) = ☺$

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

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

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

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

2.63 PRZYKŁAD – Wartościowanie $\mathfrak{v}_2$

- $\mathfrak{v}_1(\text{”x”}) = ☺$
- $\mathfrak{v}_1(\text{”y”}) = ☹$
- $\mathfrak{v}_1(\text{”z”}) = ☻$

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

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

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

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

2.66 PRZYKŁAD – Model $\mathfrak{M}_2$

1. $\mathbb{D} = \{$☺, ☹, ☻$\}$
2. $\mathbb{I}$("Donald") = ☺
3. $\mathbb{I}$("Jarosław") = ☹
4. $\mathbb{I}$("jest przyjacielem")=$\mathbb{D} \times \mathbb{D}$

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

- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Donald jest przyjacielem Jarosław"
- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Donald jest przyjacielem Donald"
- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Jarosław jest przyjacielem Donald"
- $\mathfrak{M}_2, \mathfrak{v}_1 \models$ "Jarosław jest przyjacielem Jarosław"

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

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

2.69 PRZYKŁAD – Model $\mathfrak{M}_3$

1. $\mathbb{D} = \{☺, ☹, ☻\}$
2. $\mathbb{I}$(”Donald”) = ☺
3. $\mathbb{I}$(”Jarosław”) = ☹
4. $\mathbb{I}$(”jest przyjacielem”)=$\emptyset$

2.70 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”

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

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

2.72 Metalogiczne własności $\mathbb{WRP}$

2.73 Metalogiczne własności $\mathbb{WRP}$

1. $\mathbb{WRP}$ jest niesprzeczny
2. $\mathbb{WRP}$ jest zupełny
3. $\mathbb{WRP}$ nie jest rozstrzygalny

2.74 Systemy automatycznego dowodzenia twierdzeń

2.75 Prover9/Mace4

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

2.76 Język $\mathbb{WRP}$ w Prover9/Mace4

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

$$< zero\_digit >::= "0" \tag{41}$$
$$< nonzero\_digit >::= "1"|"2"|"4"|"5"|"6"|"7"|"8"|"9" \tag{42}$$
$$< digit >::=< zero\_digit > | < nonzero\_digit > \tag{43}$$
$$< index >::=< digit > | < nonzero\_digit >< index > \tag{44}$$
$$< const >::= ("a"|"b"|\ldots|"t")[< index >] \tag{45}$$
$$< var >::= ("u"|"v"|"x"|"y"|"z")[< index >] \tag{46}$$
$$< \alpha >::=< const > | < var > \tag{47}$$
$$< seq >::=< \alpha > | < \alpha > "," < seq > \tag{48}$$
$$< pred >::= "A"|"B"|\ldots|"Z")[< index >] \tag{49}$$

2.77 Język $\mathbb{WRP}$ w Prover9/Mace4 (2)

$$< atom >::=< pred > "(" < seq > ")." \tag{50}$$
$$\begin{aligned} < \phi >::=< atom > | \\ "all" < var > \phi"."| \\ "exists" < var > \phi"."| \\ "-" < \phi > "."| \\ < \phi > "\&" < \phi > "."| \\ < \phi > "|" < \phi > "."| \\ < \phi > "->" < \phi > "."| \\ < \phi > "<->" < \phi > "." \end{aligned} \tag{51}$$

2.78 Dowodzenie w Prover9/Mace4

1. $\forall x\ A(x) \to \exists x\ A(x)$
2. $A(x) \to \exists x\ A(x)$
3. $\exists x \forall y\ P(x,y) \to \forall y \exists x\ P(x,y)$
4. $\forall x\ A(x) \lor \forall x\ B(x) \to \forall x\ (A(x) \lor B(x))$
5. $$\frac{\begin{array}{c}\forall x \forall y \forall z[P(y,z) \land P(x,y) \to P(x,z)] \\ \forall x \forall y[Q(x,y) \to P(y,x)] \\ \forall x \forall y[P(x,y) \to Q(y,x)]\end{array}}{\forall x \forall y \forall z[Q(y,z) \land Q(x,y) \to Q(x,z)]}$$

2.79 Szukanie modeli w Prover9/Mace4

1. $\forall y \exists x\ P(x,y) \to \exists x \forall y\ P(x,y)$
2. $\forall x\ (A(x) \lor B(x)) \to \forall x\ A(x) \lor \forall x\ B(x)$

2.80 Inne systemy automatycznego dowodzenia twierdzeń

`http://tptp.cs.miami.edu/`

3 Logiki opisowe

3.1 Logiki opisowe

- *Description Logics* - [?]
- geneza:
  - koniec lat 80-ych XX w.

  - sieci semantyczne – reprezentacja struktury wypowiedzi w grafach

- rodzina logik
- rozstrzygalne fragmenty $\mathbb{WRP}$
  - czasami (optymalnie): liczba kroków w rezolucji wyrażenia zależy wykładniczo od liczby predykatów w wyrażeniu
- "brak" aksjomatów
- zastosowania: Sieć Semantyczna

3.2 Sieć Semantyczna a ontologia inżynieryjna

> 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.

3.3 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 indywiduów może być wyprowadzona automatycznie. W szczególności, możemy "dowiadywać się" o relacji subsumpcji pomiędzy pojęciami.

3.4 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.

3.5 Logika opisowa $\mathcal{ALC}$

3.6 Logika opisowa $\mathcal{ALC}$

- język
- model
- tablice semantyczne

3.7 Alfabet $\mathcal{ALC}$

Definicja 12 *Alfabet $\mathcal{ALC}$ zawiera:*

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

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

   - *dwuargumentowych (*ról*):*

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

3. *funktory (nazwotwórcze od argumentów nazwowych):* $-$, $\sqcap$*;*

4. *"kwantyfikatory": $\exists.$, $\forall.$;*

5. *funktory (zdaniotwórcze od argumentów nazwowych): $\sqsubseteq$, $=$;*

6. *nawiasy: (,).*

3.8 "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

3.9 Zbiór pojęć logiki $\mathcal{ALC}$

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

Definicja 13 *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.*

3.10 Zbiór pojęć logiki $\mathcal{ALC}$ - przykłady

$HumanBeing$, $Female$, $Polish$, $\top$

$$Female \sqcap Polish$$

$$-HumanBeing$$

$$Female \sqcap HumanBeing$$

$$-\top$$

$$-(Female \sqcap Polish)$$

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

$$\exists hasChild.\top$$

$$\forall hasHair. - \top$$

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

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

3.12 Język logiki $\mathcal{ALC}$

Definicja 14 *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 rolą, 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.

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

$$Male \sqsubseteq Animal$$

$$Female \sqsubseteq -Animal$$

$$Man = HumanBeing \sqcap -Female$$

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

$$Bachelor = Man \sqcap \forall hasSpouse. - \top$$

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

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

3.14 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*)

$$\top \sqsubseteq \exists hasShadow.\top$$

(zainspirowany przez J. Pogonowskiego)

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

$$hasChild \sqsubseteq hasDescendant$$

$$hasWife = hasHusband$$

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

$HumanBeing(Jan)$
$\top(Jan)$

$$(Female \sqcap Polish)(Jan)$$

$$(-HumanBeing)(Burek)$$

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

3.17 Czego nie można powiedzieć w $\mathcal{ALC}$?

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

3.18 Modele $\mathcal{ALC}$

Definicja 15 *Parę* $\mathfrak{M} =< \mathbb{D}, \mathbb{I} >$ *nazywamy modelem dla języka* $\mathcal{ALC}$ *jeżeli:*

1. $\mathbb{D} \neq \emptyset$,
2. $\mathbb{I}$ *jest taką funkcją, która:*
   (a) *odwzorowuje zbiór* $\mathbb{CONST}$ *w zbiór* $\mathbb{D}$,
   (b) *odwzorowuje zbiór pojęć w* $\wp(\mathbb{D})$, *przy czym spełnione są następujące warunki:*
$$\mathbb{I}(\top) = \mathbb{D}$$
$$\mathbb{I}(-\delta) = \mathbb{D} \setminus \mathbb{I}(\delta)$$
$$\mathbb{I}(\delta_1 \sqcap \delta_2) = \mathbb{I}(\delta_1) \cap \mathbb{I}(\delta_2)$$
$$\mathbb{I}(\exists \rho.\delta) = \{x \in \mathbb{D} : \exists y < x, y >\in \mathbb{I}(\rho) \wedge y \in \mathbb{I}(\delta)]\}$$
$$\mathbb{I}(\forall \rho.\delta) = \{x \in \mathbb{D} : \forall y[< x, y >\in \mathbb{I}(\rho) \rightarrow y \in \mathbb{I}(\delta)]\}$$
   (c) *odwzorowuje zbiór ról w* $\wp(\mathbb{D} \times \mathbb{D})$.

3.19 Spełnianie w $\mathcal{ALC}$

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

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

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

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

gdzie: ϵ_1, ϵ_2 reprezentują nazwy pojęć *lub* ról.

3.20 Spełnianie w $\mathcal{ALC}$ (2)

$$\models \phi \equiv \forall \mathfrak{M}\ \mathfrak{M} \models \phi. \tag{56}$$

3.21 Spełnianie w $\mathcal{ALC}$ (3)

$$\mathfrak{M} \models \Delta \equiv \forall \phi \in \Delta\ \mathfrak{M} \models \phi. \tag{57}$$

3.22 Spełnianie w $\mathcal{ALC}$ (4)

$$\begin{aligned} &\delta \text{ jest spełnialne ze względu na pudełko terminologiczne } \Delta \equiv \\ &\quad \text{istnieje taki model } \mathfrak{M} =< \mathbb{D}, \mathbb{I} > \text{ że, } \mathfrak{M} \models \Delta \text{ oraz } \mathbb{I}(\delta) \neq \emptyset. \end{aligned} \tag{58}$$

3.23 Spełnianie w $\mathcal{ALC}$ (5)

Fakt 1 $\models \delta_1 \sqsubseteq \delta_2$ *wtedy i tylko wtedy, gdy dla dowolnego pudełka terminologicznego* Δ, $\delta_1 \sqcap -\delta_1$ *nie jest spełnialne ze względu na* Δ.

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

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

1. Utwórz pojęcie $\delta_1 \sqcap -\delta_2$
2. Wstaw $(\delta_1 \sqcap -\delta_2)(\alpha)$ jako korzeń tablicy analitycznej T (dla dowolnej stałej α)!

3. Korzystając z atomowych tablic analitycznych zredukuj T!

4. Czy T jest zamknięta?

   - TAK, $\delta_1 \sqsubseteq \delta_2$ jest prawem logiki.
   - NIE, $\delta_1 \sqsubseteq \delta_2$ nie jest prawem logiki.

3.25 Zamknięte i otwarte tablice analityczne w logikach opisowych

Definicja 16 *Tablice analityczna jest* zamknięta*, jeżeli każda jej gałąź:*

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

2. *lub zawiera wyrażenie* $(-\top)(\alpha)$.

Tablice analityczna jest otwarta*, jeżeli nie jest zamknięta.*

3.26 Tablice analityczne w $\mathcal{ALC}$

- $\mathbb{TON}$:

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

3.27 Tablice analityczne w $\mathcal{ALC}$ (2)

3.28 Tablice analityczne w $\mathcal{ALC}$ (3)

- $\mathbb{TO}\exists$:

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

  gdzie: β jest nową stałą.

- $\mathbb{TO}\forall$:

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

  gdzie: β jest dowolną stałą.

3.29 Tablice analityczne w $\mathcal{ALC}$ (4)

- $\mathbb{TON}\exists$: $$\begin{array}{c}\boldsymbol{(-\exists R.\delta)(\alpha)}] \\ | \\ (\forall R.-\delta)(\alpha)\end{array}$$
- $\mathbb{TON}\forall$: $$\begin{array}{c}\boldsymbol{(-\forall R.\delta)(\alpha)}] \\ | \\ (\exists R.-\delta)(\alpha)\end{array}$$

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

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

1. Utwórz pojęcia: $\delta_1 \sqcap -\delta_2$ oraz $\delta_2 \sqcap -\delta_1$
2. Wstaw $(\delta_1 \sqcap -\delta_2)(\alpha)$ i $(\delta_2 \sqcap -\delta_1)(\alpha)$ 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.

3.31 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.

3.32 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?

3.33 Reprezentacja relacji w logice

3.34 Relacje w życiu i w logice

3.35 Reprezentacja relacji w $\mathbb{WRP}$

- $P(x, y)$:
  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. ...

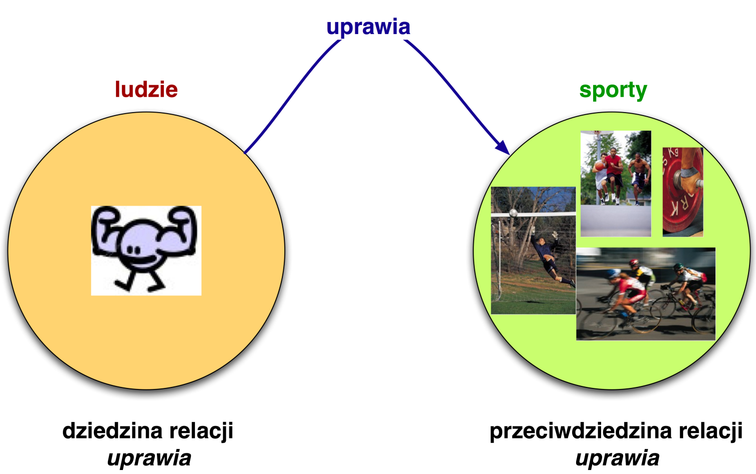
3.36 Reprezentacja relacji w modelach $\mathbb{WRP}$

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

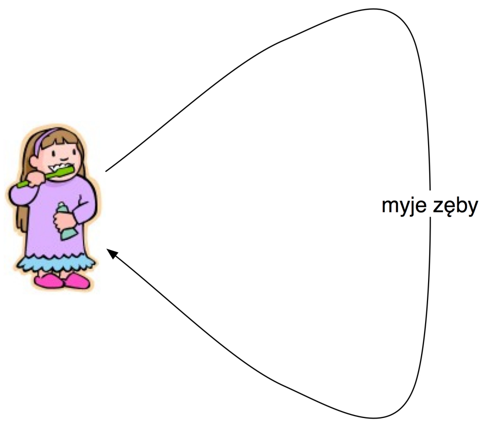
Tabela 4: Przykłady podstawiania w $\mathbb{PZW}_{\mathbb{WRP}}$

$\mathbb{PZW}_{\mathbb{KRZ}}$	**podstawienie**	**wynik podstawienia**
$P(x,y) \to R(y,x)$	x/y	$P(y,y) \to R(y,y)$
$P(x,y) \to R(y,x)$	x/a	$P(a,y) \to R(y,a)$
$\forall x\ P(x,y) \to R(y,x)$	x/b	$\forall x\ P(x,y) \to R(y,b)$
$\exists y\ P(x,y)$	x/y	-
$\exists y\ P(x,y)$	x/c	$\exists y\ P(c,y)$

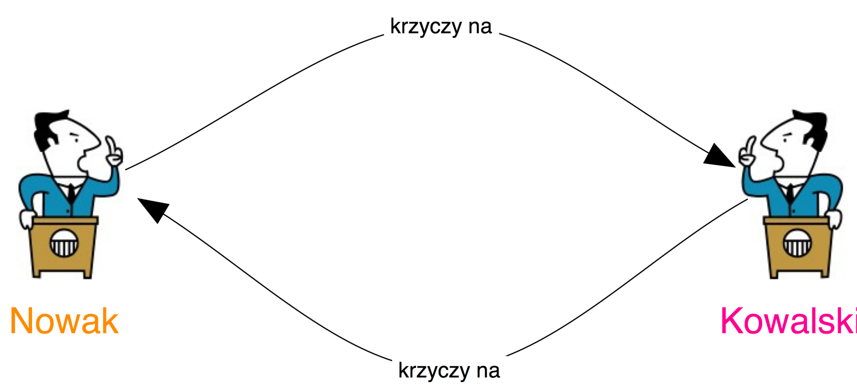
miłość	x kocha y	Jan kocha Annę.
		Tomasz kocha Ewę.
		Tomasz kocha Annę.
		Tomasz kocha Jana.
		...
ojcostwo	x spłodził y	Jan spłodził Zofię.
		...
bycie większym	x jest większe niż y	...
bycie dłużnikiem	x jest dłużnikiem y	...
pożyczanie pieniędzy	x pożyczył od y z zł	Jan pożyczył od Tomasza 100 zł.



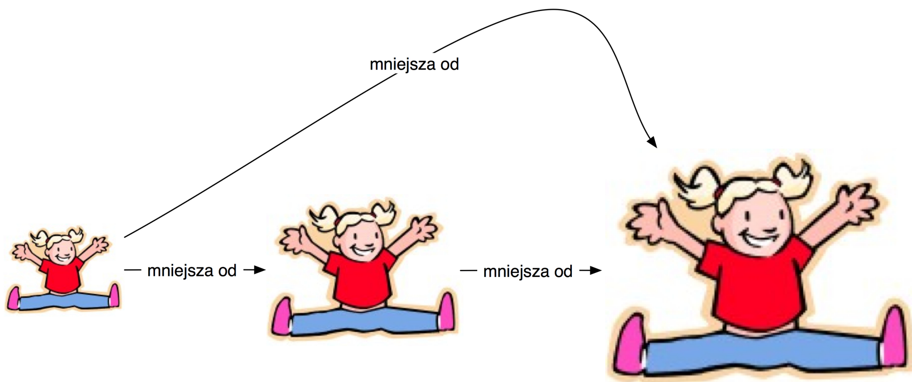
Rysunek 1: Dziedzina i przeciwdziedzina relacji



Rysunek 2: Zwrotność



Rysunek 3: Symetryczność



Rysunek 4: Przechodność

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

3.37 Obrazkowa teoria relacji

¡1¿ ¡2¿ ¡3¿ ¡4¿

3.38 Dziedzina i przeciwdziedzina relacji

Definicja 17 *Jeżeli R jest relacją dwuargumentową, to x należy do* dziedziny *relacji R wtedy i tylko wtedy, gdy*

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

Definicja 18 *Jeżeli R jest relacją dwuargumentową, to x należy do* przeciwdziedziny *relacji R wtedy i tylko wtedy, gdy*

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

3.39 Dziedzina i przeciwdziedzina relacji - przykłady

3.40 (Najważniejsze) rodzaje relacji dwuargumentowych

1. zwrotne i przeciwzwrotne
2. symetryczne, asymetryczne, antysymetryczne
3. przechodnie
4. funkcje

3.41 Relacje zwrotne

Definicja 19 *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

3.42 Relacje przeciwzwrotne

Definicja 20 *Relacja dwuargumentowa R jest* przeciwzwrotna *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

3.43 Relacje symetryczne

Definicja 21 *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

3.44 Relacje asymetryczne

Definicja 22 *Relacja dwuargumentowa R jest* asymetryczna *wtedy i tylko wtedy, gdy*

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

przykłady:

1. $<$
2. bycie dłużnikiem

antyprzykłady:

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

3.45 Relacje antysymetryczne

Definicja 23 *Relacja dwuargumentowa R jest* antysymetryczna *wtedy i tylko wtedy, gdy*

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

przykłady:

1. $\leq$

antyprzykłady:

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

3.46 Relacje przechodnie

Definicja 24 *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

3.47 Funkcje

Definicja 25 *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

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

3.48 Trzyczynnikowa klasyfikacja relacji - przykłady

3.49 Relacje (ściśle) porządkujące

Definicja 26 *Relacja jest ściśle porządkująca (jest porządkiem) jeżeli jest asymetryczna i przechodnia.*

przykłady:

1. $<$
2. być głupszym niż

antyprzykłady:

1. równoległość (prostych)
2. $\leq$

3.50 Relacje równoważnościowe

Definicja 27 *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

3.51 Operacje na relacjach

- ogólne (teoriomnogościowe): suma, iloczyn, różnica, dopełnienie, itd.

- lokalne:

  1. inwers relacji
  2. iloczyn względny relacji

3.52 Iloczyn względny relacji

Definicja 28 *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 $<$

3.53 Inwers relacji

Definicja 29 *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

3.54 Logika opisowa $\mathcal{SROIQ}$

3.55 Jaka logika opisowa jest najlepsza i dlaczego jest to $\mathcal{SROIQ}$?

- [?]
- 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
    - $\exists R.\mathtt{Self}$ - (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 ($^-$)
- (wieloargumentowy) iloczyn względny ról ($\circ$)

- negacja (przy-)zdaniowa: "$\sim$"

3.56 Jaka logika opisowa jest najlepsza i dlaczego jest to $\mathcal{SROIQ}$? (2)

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

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

- Każdy kij ma dwa końce.

$$Stick \;\sqsubseteq\; (\geq 2 hasEnd.\top) \sqcap (\leq 3 hasEnd.\top)$$

- Każdy, kto zbudował dom, posadził też drzewo.

$$\exists build.House \sqsubseteq \exists plant.Tree$$

- Przyjaciele moich przyjaciół są moimi przyjaciółmi.

$$hasFriend \circ hasFriend \sqsubseteq hasFriend$$

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

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

3.59 Alfabet $\mathcal{SROIQ}$ – DEFINICJA

Alfabet $\mathcal{SROIQ}$ zawiera:

1. zbiór stałych indywidualnych $\mathbb{CONST} := \{a_1, a_2, \dots\}$,
2. zbiór predykatów
   - jednoargumentowych (pojęć atomicznych):
   $$\mathbb{CONCEPT} := \{\top, A_1, A_2, \dots\}$$
   - dwuargumentowych (ról atomicznych):
   $$\mathbb{ROLE} := \{\mathtt{U}, =, R_1, R_2, \dots\}$$
3. funktory (nazwotwórcze od argumentów nazwowych): $\neg$, $\sqcap$, $\sqcup$, $\circ$, $^-$, $\{\dots\}$;
4. "kwantyfikatory": $\exists.$, $\forall.$, $\geq n.$, $\leq n.$, $\exists.\mathtt{Self}$
5. funktory (zdaniotwórcze od argumentów nazwowych): $\sqsubseteq$, $\equiv$, $\mathtt{Dis}$, $=$, $\mathtt{Refl}$, $\mathtt{Irrefl}$, $\mathtt{Sym}$, $\mathtt{Asym}$, $\mathtt{Trans}$
6. funktor (zdaniotwórczy od argumentu zdaniowego): $\sim$;
7. nawiasy: (,).

3.60 Zbiór pojęć logiki $\mathcal{SROIQ}$ – przykłady

$$Tailor \sqcap Sailor$$

$$Tailor \sqcup Sailor$$

$$\geq 2 hasEnd.\top$$

$$\leq 3 hasEnd.\top$$

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

$$\{John\}$$

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

$$\exists likes.\mathtt{Self}$$

3.61 Zbiór pojęć logiki $\mathcal{SROIQ}$

Definicja 30 *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 $\exists\rho.\mathtt{Self}$ jest pojęciem.*

3.62 Zbiór ról logiki $\mathcal{SROIQ}$ – przykłady

$$hasChild$$

$$\mathtt{U}$$

$$hasChild^-$$

$$hasChild \circ hasChild$$

3.63 Zbiór ról logiki $\mathcal{SROIQ}$

Definicja 31 *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 $\rho_1, \rho_2, \ldots, \rho_n$ są rolami różnymi od roli uniwersalnej $\mathtt{U}$, to $\rho_1 \circ \rho_2 \circ \cdots \circ \rho_n$ jest rolą (łańcuchem ról).*

3.64 Język logiki $\mathcal{SROIQ}$

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

Definicja 32 *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:*
   (a) *hierarchia ról jest $<$-regularna (dla pewnego $<$),*
   (b) *zbiór asercji ról jest prosty.*

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

3.65 Pudełko terminologiczne $\mathcal{SROIQ}$

Definicja 33 *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.*

3.66 Pudełko opisowe $\mathcal{SROIQ}$

Definicja 34 *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.*

3.67 Pudełko relacyjne $\mathcal{SROIQ}$

Definicja 35 *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 $\mathtt{Dis}(\rho_1, \rho_2)$ należy do pudełka relacyjnego;*
3. *Jeżeli ρ jest rolą logiki $\mathcal{SROIQ}$, to $\mathtt{Irrefl}(\rho)$, $\mathtt{Asym}(\rho)$ należą do pudełka relacyjnego.*
4. *Jeżeli ρ jest rolą logiki $\mathcal{SROIQ}$, to $\mathtt{Refl}(\rho)$, $\mathtt{Sym}(\rho)$ i $\mathtt{Trans}(\rho)$ należą do pudełka relacyjnego.*

Zbiór wyrażeń spełniających warunek **??** nazywamy *hierarchią ról*.
Zbiór wyrażeń spełniających warunki **??** lub **??** nazywamy *zbiorem asercji ról*.

3.68 Regularny porządek ról

Definicja 36 *Niech $\mathcal{R}$ będzie zbiorem ról logiki $\mathcal{SROIQ}$. Przeciwzwrotna i przechodnia relacja $<$ w zbiorze $\mathcal{R}$ jest* regularnym porządkiem*, gdy dla każdych $\rho_1, \rho_2 \in \mathcal{R}$*

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

3.69 Regularne aksjomaty ról

Definicja 37 *Niech $\rho_1 \circ \rho_2 \circ \cdots \circ \rho_n$ będzie rolą logiki $\mathcal{SROIQ}$ taką że żadna z ról $\rho_1, \rho_2, \ldots, \rho_n$ nie jest rolą uniwersalną. Aksjomat subsumpcji ról $\rho_1 \circ \rho_2 \circ \cdots \circ \rho_n \sqsubseteq \rho$ ($\rho \neq \mathtt{U}$) 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* $1 \leq i < n$*)*

3.70 Regularne zbiory aksjomatów ról

Definicja 38 *Hierarchia ról jest* $<$*-regularna jeżeli każdy aksjomat w tej hierarchii jest* $<$*-regularny.*

3.71 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 $\sqsubseteq_{\mathcal{A}}$.

3.72 Role proste (2)

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

3.73 Role proste (3)

Definicja 39 *Rola* ρ *jest prosta ze względu na zbiór* $\mathcal{A}$ *aksjomatów subsumpcji ról, gdy* $\rho_1 \circ \rho_2 \circ \cdots \circ \rho_n \sqsubseteq^*_{\mathcal{A}^-} \rho$ *implikuje, że* $n = 1$*.*

3.74 Role proste (2)

Definicja 40 *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.*

3.75 Role proste (3)

Definicja 41 *Zbiór asercji ról jest prosty gdy wszystkie aksjomaty występujące w nim o postaci* $\texttt{Irrefl}(\rho)$, $\texttt{Asym}(\rho)$ *lub* $\texttt{Dis}(\rho_1, \rho_2)$, *zawierają tylko role proste.*

4 Bibliografia