

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

PUTÁ MEDZI FORMÁLNYMI KONTEXTAMI

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

PUTÁ MEDZI FORMÁLNYMI KONTEXTAMI

DIPLOMOVÁ PRÁCA

Študijný program:

Informatika

Pracovisko:

Ústav informatiky

Vedúci diplomovej práce:

RNDr. Ondrej Krídlo, PhD.

Košice 2013

Bc. Patrik MIHALČIN

Zadanie záverečnej práce



Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Patrik Mihalčín
Študijný program: Informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: Diplomová práca
Jazyk záverečnej práce: slovenský

Názov: Putá medzi formálnymi kontextami

Cieľ: Cieľom diplomanta je priblížiť sa riešeniu otvoreného problému "jednoduchej konštrukcie formálneho kontextu, ktorého konceptový zväz je izomorfný so zväzom všetkých pút medzi danými kontextami". Približným riešením môže byť akákoľvek aj zložitá konštrukcia, rozšírenie problému do iných ako klasických logík, alebo teoreticky popis zložitosti spomínaného riešenia. Ďalším pokračovaním bádania v tejto diplomovej práci je vyhľadávanie tzv. "zmysluplných" pút, resp. pút, ktoré zodpovedajú nejakému implicitnému, resp. dodatočne pridanému vzťahu skúmaných dát.

Literatúra:

1. Bělohlávek R., Lattices of fixed points of fuzzy galois connections. Mathematical Logic Quartely.
2. Ganter B., Wille R., Formal Concept Analysis, Mathematical Foundations, Berlin, Germany, Springer Verlag 1999
3. Křídlo O, Ojeda-Aciego M., Krajčí S., Bonds vs Extents of Direct Product od Two Formal Contexts, Proceedings of CLA 2010, Sevilla, Spain 2010

Vedúci: RNDr. Ondrej Křídlo, PhD.
Ústav: ÚINF - Ústav informatiky
Riaditeľ ústavu: prof. RNDr. Viliam Geffert, DrSc.
Dátum zadania: 27.03.2012

Dátum schválenia: 22.04.2013


prof. RNDr. Viliam Geffert, DrSc.
riaditeľ ústavu



Vyhlásenie

Vyhlasujem, že túto diplomovú prácu som vypracoval samostatne na základe štúdiom získaných poznatkov. Použitú literatúru som uviedol v práci.

Košice, apríl 2013

Patrik Mihalčín

Pod'akovanie

Ďakujem RNDr. Ondrejovi Krídlovi, PhD. za odborné vedenie, inšpirujúce podnety, cenné rady a pripomienky.

Abstrakt

V tejto práci sa zaoberáme interkontextovou štruktúrou úplného zväzu tzv. pút medzi dvoma fuzzy formálnymi kontextami. Naša práca sa v úvode zaoberá základnými znalosťami formálnej konceptovej analýzy a jej fuzzy rozšírením ako aj efektívnym vyhľadávaním všetkých konceptov akéhokoľvek fuzzy formálneho kontextu. Daný efektívny prístup sme ako jeden z cieľov práce rozšírili pre potreby vyhľadávania všetkých pút medzi dvoma formálnymi kontextami a na jeho základe sme poskytli návrh pseudoriešenia jedného z otvorených problémov formálnej konceptovej analýzy. V práci sa zaoberáme aj ďalším jednoduchším prístupom vyhľadávania pút dvoch kontextov, ktorého nevýhodou je neúplnosť výstupu. V závere práce je uvedené možné využitie pút v praxi.

Kľúčové slová: fuzzy formálny kontext, konceptový zväz, putá, priamy súčin dvoch kontextov

Abstract

In this thesis we deal with intercontextual structure called a complete lattice of bonds between two fuzzy formal contexts. This thesis is organized as follows. At the beginning we present preliminaries from formal concept analysis and its fuzzy extension as well as a fast bottom-up algorithm to compute all concepts of arbitrary fuzzy formal context. One of our objectives was extension of this algorithm in order to compute all bonds between two formal contexts and after that we provided proposal of pseudosolution which contributes to one of open problems in formal concept analysis. We also deal with another approach of computing bonds between two formal contexts, but the disadvantage of it is incomplete output. The last section is dedicated to possible use of bonds in practice.

Keywords: fuzzy formal context, concept lattice, bonds, direct product of two formal contexts

Obsah

Obsah	7
Zoznam ilustrácií	9
Zoznam tabuliek	10
Úvod	11
1 Základy.....	13
1.1 Štruktúry, s ktorými pracujeme	13
1.1.1 Posety.....	13
1.1.2 Zväzy.....	14
1.1.3 Galoisove konexie.....	15
1.2 Klasická formálna konceptová analýza	17
1.3 Fuzzy formálna konceptová analýza	22
1.3.1 Fuzzy množina	23
1.3.2 Lukasiewiczova logika.....	23
1.3.3 Fuzzy uzáverový operátor.....	24
1.3.4 Formálna konceptová analýza dát s fuzzy atribútmi.....	25
2 Ako hľadať koncepty v kontexte	29
2.1 Problém: nájsť zväz všetkých fixpointov	29
2.1.1 Predbežné úvahy	29
2.1.2 Spojitosť medzi uzáverovými operátormi a konceptovými zväzmi	29
2.1.3 Cieľ algoritmu.....	30
2.2 Horní susedia a ich generátory	30
2.3 Algoritmus pre hľadanie všetkých fixpointov	32
2.4 Príklad výpočtu.....	34
3 Ako hľadať putá medzi dvoma kontextami	37
3.1 Multifunkcia, bond	37
3.2 Horní susedia a ich generátory	39
3.2.1 Ľavý a pravý uzáver multifunkcie	39
3.2.2 Generátory horných susedov.....	40
3.2.3 Generovanie horných susedov	42
3.3 Algoritmus pre hľadanie všetkých pút	44
3.4 Príklad výpočtu.....	45
3.5 Porovnanie nášho algoritmu s brute-force prístupom	51

4	Pseudoriešenie 9. otvoreného problému.....	53
4.1	Transformácia zväzu pút na prirodzene malý kontext	53
5	Spojitosť medzi putami a extentami priameho súčinu dvoch formálnych kontextov	57
6	Využitie pút.....	60
	Záver	63
	Zoznam použitej literatúry	64
	Prílohy	65

Zoznam ilustrácií

Obr. 1 Hasseho čiarový diagram pre úplný zväz konceptov kontextu v Tab. 1	22
Obr. 2 Fuzzy konceptový zväz fuzzy kontextu Tab. 6	27
Obr. 3 Fuzzy koncept ako maximálny obdĺžnik v relácii I.....	27
Obr. 4 Priebeh funkcie NEIGHBORS pre najmenší fixpoint nášho príkladu	35
Obr. 5 Úplný zväz fixpointov	36
Obr. 6 Puto medzi dvoma kontextami	38
Obr. 7 Ľavý uzáver L-bondu β	40
Obr. 8 Úplný zväz intentov druhého L-kontextu	46
Obr. 9 Úplný zväz extentov prvého L-kontextu.....	47
Obr. 10 Objekty a atribúty L-bondu skúmané vo funkcii NEIGHBORS	48
Obr. 11 Priebeh funkcie MOVES1 pre najmenší L-bond a prvý objekt priestoru X_1	49
Obr. 12 Priebeh časti funkcie NEIGHBORS, kde vyberáme reálnych horných susedov L-bondu β	50
Obr. 13 Úplný zväz všetkých L-bondov	51
Obr. 14 Závislosť času behu algoritmov na veľkosti hľadaného zväzu pút	52
Obr. 15 Zväz všetkých L -bondov a zvýraznené všetky infimum a suprérum ireducibilné prvky	55
Obr. 16 Úplný zväz extentov prirodzene malého kontextu	56
Obr. 17 Priamy súčin kontextov C_1 a C_2	58
Obr. 18 Úplný zväz všetkých pút a zvýraznené putá nájdené pomocou priameho súčinu	59
Obr. 19 Základný kontext preferencií študentov a externé kontexty študijných výsledkov a požiadaviek univerzít.....	61
Obr. 20 Vytvorenie puta medzi objektovým a atribútovým kontextom	62

Zoznam tabuliek

Tab. 1 Príklad objektovo-atribútového modelu	18
Tab. 2 Príklad použitia zobrazenia $\uparrow$	20
Tab. 3 Príklad použitia zobrazenia $\downarrow$	20
Tab. 4 Lukasiewiczova silná konjunkcia $\otimes$	24
Tab. 5 Lukasiewiczova implikácia $\rightarrow$	24
Tab. 6 Výsledky študentov v okruhoch predmetov	25
Tab. 7 Ilustračný L-kontext	34
Tab. 8 Prvý L-kontext ilustračného príkladu	46
Tab. 9 Druhý L-kontext ilustračného príkladu	46
Tab. 10 Prirodzene malý kontext	55

Úvod

Formálna konceptová analýza je jednou z data-miningových metód, ktorej základné piliere postavili nemeckí matematici Bernhard Ganter a Rudolf Wille prácou Formal Concept Analysis [1] a stala sa veľmi užitočným nástrojom pre charakterizovanie dát s objektovo-atribútovým charakterom. Povinný základ formálnej konceptovej analýzy si predstavíme začínajúc kapitolou 1, kde sme poznatky čerpali z už spomínanej práce Formal Concept Analysis [1]. Čoskoro po predstavení „klasickej“ formálnej konceptovej analýzy boli uvedené prístupy na jej zovšeobecnenie použitím myšlienok fuzzy teórie množín. V tejto práci bude v sekcii 1.3 stručne predstavené fuzzy rozšírenie formálnej konceptovej analýzy Bělohlávkom uvedené v článkoch [7], [8], t.j. základné pojmy formálnej konceptovej analýzy ako sú kontext a koncept.

V kapitole 2 sa zaoberáme efektívnym vyhľadávaním konceptov v kontexte a podrobne sa čitateľovi predostiera myšlienka hľadania všetkých uzavretých množín, tzv. fixpointov algoritmom, ktorý využíva paradigmy horných susedov. Tento algoritmus sme objavili v článku [2], ktorý vytvoril tím okolo Bělohlávka. Skryté významy nájdené v kontexte sa nazývajú koncepty a ak si zoberieme do úvahy dva kontexty, tak skryté významy medzi týmito kontextami nazveme putá medzi kontextami, ktorými vieme vyjadriť vlastnosti medzikontextových vzťahov. V roku 2006 boli v Drážďanoch prednesené a zoskupené v zbierke [5] otvorené problémy formálnej konceptovej analýzy a jedným z nich je problém týkajúci sa chápania pút medzi kontextami. Deviaty problém hľadá odpoveď na otázku hľadania prirodzene malého kontextu pre uzavretý systém pút medzi kontextami alebo inak povedané hľadania jednoduchšej konštrukcie kontextu, ktorého konceptový zväz je izomorfný so zväzom všetkých pút medzi danými kontextami. Táto skutočnosť bola motiváciou nášho výskumu, kde sa snažíme priblížiť k riešeniu tohto problému. Poznatky o putách sme čerpali hlavne z článkov [3] od Krídla, Krajčího a Ojeda-Aciego, [6] od Krötzscha, Hitzlera a Zhanga a knihy [1] od Gantera a Willeho.

V kapitole 3 popisujeme náš prínos, kde začíname vysvetlením základných pojmov, napr. puto medzi kontextami, pokračujeme rozšírením Bělohlávkovho algoritmu pre hľadanie konceptového zväzu spolu s usporiadaním uvedeného v článku [2] na algoritmus pre hľadanie zväzu pút spolu s usporiadaním a končíme názorným príkladom výpočtu nášho algoritmu. Následne v kapitole 4 popisujeme transformáciu zväzu pút na

prirrodzene malý kontext, ktorého konceptový zväz je izomorfný s úplným zväzom všetkých pút medzi vstupnými kontextami. Takýto prístup nie je ideálny kvôli tomu, že naše riešenie nie je priame v zmysle, že pre dané 2 kontexty vrátíme jednoduchým spôsobom prirrodzene malý kontext. Najprv si potrebujeme vybudovať zväz pút a až ten následne pretransformujeme na prirrodzene malý kontext pomocou známych faktov teórie zväzov týkajúcich sa suprérum/infimum ireducibilných prvkov. Naše riešenie nazveme pseudoriešením kvôli tomuto nedostatku.

Kapitola 5 je venovaná ďalšiemu spôsobu hľadania pút medzi dvoma kontextami prostredníctvom priameho súčinu dvoch kontextov, pričom všetky detaily sú obsiahnuté v článku [4].

V kapitole 6 aplikujeme nadobudnuté znalosti v novom prístupe hľadania skrytých informácií vo viacerých spolu súvisiacich kontextoch. Tento prístup popíšeme na modelovom príklade, kde sa snažíme vypomôcť s identifikovaním správnych kandidátov pre dané univerzity a správnych univerzít pre daných študentov, pričom našou snahou je čím väčšia spravodlivosť rozdelenia.

Zámerom práce je podať čitateľovi dostatočne deskriptívny pohľad na putá medzi kontextami a ukázať ich význam a rolu vo svete formálnej konceptovej analýzy. Všetky spomenuté algoritmy boli úspešne naimplementované a otestované v programovacom jazyku Java.

1 Základy

V tejto kapitole sa čitateľ oboznámi so všetkými potrebnými znalosťami, ktoré sú základnými stavebnými blokmi pre náš výskum. Ak je čitateľ familiárny so znalosťami ako posety, úplné zväzy, Galoisove konexie, klasická formálna konceptová analýza a formálna konceptová analýza dát s fuzzy atribútmi môže spokojne túto kapitolu preskočiť.

1.1 Štruktúry, s ktorými pracujeme

Táto podkapitola uvádza potrebné znalosti z teórie úplných zväzov, ktoré sú základom formálnej konceptovej analýzy. Všetky znalosti boli čerpané z práce Formal Concept Analysis od Bernharda Gantera a Rudolfa Willeho [1].

1.1.1 Posety

Definícia 1.1. Uvažujme množinu X a binárnu reláciu $R \subseteq X \times X$, ktorá je na X

1. reflexívna, čiže $(\forall x \in X)(x, x) \in R$,
2. antisymetrická, čiže $(\forall x, y \in X) \left(((x, y) \in R, x \neq y) \Rightarrow (y, x) \notin R \right)$,
3. tranzitívna, čiže $(\forall x, y, z \in X) \left(((x, y) \in R, (y, z) \in R) \Rightarrow (x, z) \in R \right)$.

Reláciu R nazývame **čiasťočným usporiadaním** na množine X . Množinu X nazývame **nosič** usporiadania a dvojicu (X, R) nazývame čiastočne usporiadanou množinou, teda **posetom**.

Definícia 1.2. Nech X je poset.

1. Prvok $x \in X$ nazývame **najväčším** prvkom posetu X , ak každý iný prvok z X je od neho menší.
2. Prvok $x \in X$ nazývame **najmenším** prvkom posetu X , ak každý iný prvok z X je od neho väčší.
3. Prvky $x, y \in X$ nazývame **porovnateľné**, ak $x \leq y$ alebo $y \leq x$.
4. Prvky $x, y \in X$ nazývame **neporovnateľné**, ak nie sú porovnateľné.
5. Poset X nazývame **ret'azcom**, ak sú každé jeho dva prvky porovnateľné.

Definícia 1.3. Nech X je poset a nech $P \subseteq X$. Prvok $x \in X$ nazývame **horným (dolným) ohraničením** množiny P , ak pre všetky $p \in P$ platí $p \leq x$ ($p \geq x$). Ak spomedzi všetkých horných (dolných) ohraničení množiny P existuje nejaké najmenšie (najväčšie) y , tak toto ohraničenie nazveme **suprémom (infimom)** množiny P , čo symbolicky vyjadríme ako $y = \sup P$ ($y = \inf P$).

Definícia 1.4. Hovoríme, že posety $(X, \leq_X)$ a $(Y, \leq_Y)$ sú **izomorfné** a zapisujeme $(X, \leq_X) \cong (Y, \leq_Y)$, ak existuje bijektívne zobrazenie $\gamma: X \rightarrow Y$ také, že

$$(\forall x_1, x_2 \in X) (x_1 \leq_X x_2 \Leftrightarrow \gamma(x_1) \leq_Y \gamma(x_2)).$$

Potom zobrazenie γ nazývame **izomorfizmom** medzi posetmi $(X, \leq_X)$ a $(Y, \leq_Y)$.

1.1.2 Zväzy

Definícia 1.5. Poset $(L, \leq)$ nazývame **zväz**, ak každá dvojprvková podmnožina nosiča L má suprémum aj infimum.

Vďaka existencii a jednoznačnosti suprema a infima pre každé dva prvky $a, b \in L$, môžeme na zväze L definovať nasledujúce binárne operácie:

Definícia 1.6.

1. **Priesekom** $\wedge$ nazývame binárnu operáciu typu $L \times L \rightarrow L$ definovanú predpisom $(\forall a, b \in L) a \wedge b = \inf \{a, b\}$.
2. **Spojením** $\vee$ nazývame binárnu operáciu typu $L \times L \rightarrow L$ definovanú predpisom $(\forall a, b \in L) a \vee b = \sup \{a, b\}$.

Ak zväz má najmenší, resp. najväčší prvok, zvykneme ho označovať symbolom 0, resp. symbolom 1.

Definícia 1.7. Poset $(L, \leq)$ nazývame **úplný zväz**, ak každá podmnožina nosiča L má suprémum aj infimum.

Definícia 1.8. Algebru $\langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$ nazývame **úplný reziduovaný zväz**, ak

-
1. $\langle L, \wedge, \vee, 0, 1 \rangle$ je úplný ohraničený zväz s najmenším prvkom 0 a najväčším prvkom 1
 2. $\langle L, \otimes, 1 \rangle$ je komutatívny monoid, teda $\otimes$ je binárna relácia, ktorá je komutatívna, asociatívna a pre každý prvok $a \in L$ platí $a \otimes 1 = a$
 3. $\otimes, \rightarrow$ tvoria adjungovanú dvojicu, teda pre všetky prvky $a, b, c \in L$ platí $a \otimes b \leq c$ práve vtedy, keď $a \leq b \rightarrow c$, pričom relácia $\leq$ je usporiadanie v zväze L definovaná pomocou prieseku $\wedge$ a spojenia $\vee$.

Operácie $\otimes$ (multiplikácia, t-norm) a $\rightarrow$ (reziduum) plnia úlohu fuzzy konjunkcie a fuzzy implikácie a zabezpečujú štruktúru reziduovaného zväzu na množine pravdivostných hodnôt L .

1.1.3 Galoisove konexie

Definícia 1.9. Nech $(X, \leq_X)$ a $(Y, \leq_Y)$ sú posety. Nech $\alpha: X \rightarrow Y$ a $\beta: Y \rightarrow X$ sú zobrazenia medzi týmito posetmi. Dvojicu (α, β) nazývame (antitónnou) **Galoisovou konexiou** medzi posetmi $(X, \leq_X)$ a $(Y, \leq_Y)$, ak pre každé $x, x_1, x_2 \in X, y, y_1, y_2 \in Y$ platí:

1. $x_1 \leq x_2 \Rightarrow \alpha(x_1) \geq \alpha(x_2)$,
2. $y_1 \leq y_2 \Rightarrow \beta(y_1) \geq \beta(y_2)$,
3. $x \leq \beta(\alpha(x))$,
4. $y \leq \alpha(\beta(y))$.

Potom hovoríme, že zobrazenia α, β sú (vzájomne) **duálne adjungované**. Taktiež hovoríme, že zobrazenie α je **duálnym adjunktom** zobrazenia β , a naopak, že zobrazenie β je **duálnym adjunktom** zobrazenia α .

Priamo z definície vyplývajú zaujímavé vlastnosti skladania zobrazení tvoriacich Galoisovu konexiu.

Lema 1.1. Nech X, Y sú posety, a nech (α, β) je Galoisova konexia medzi X a Y .

Potom platí

$$\alpha = \alpha \circ \beta \circ \alpha \quad \text{a} \quad \beta = \beta \circ \alpha \circ \beta.$$

Štyri podmienky „galoisovosti“ dvoch zobrazení uvedené v definícii 1.9 sú spoločne ekvivalentné jedinej podmienke.

Lema 1.2. Nech X, Y sú posety a nech $\alpha: X \rightarrow Y, \beta: Y \rightarrow X$. Potom dvojica zobrazení (α, β) je Galoisovou konexiou medzi X a Y práve vtedy, keď pre všetky $x \in X, y \in Y$ platí

$$x \leq \beta(y) \Leftrightarrow y \leq \alpha(x)$$

Lema 1.3. Nech X, Y sú posety a nech (α, β) je Galoisovou konexiou medzi X a Y . Potom, ak poset X má najmenší prvok 0_X , tak poset Y musí mať najväčší prvok 1_Y , a platí

$$\alpha(0_X) = 1_Y$$

Nasledujúce tvrdenie hovorí o tom, že každé zobrazenie medzi dvoma posetmi môže figurovať nanajvýš v jednej Galoisovej konexii.

Lema 1.4. Nech X, Y sú posety a nech $(\alpha, \beta_1), (\alpha, \beta_2)$ sú Galoisove konexie medzi X a Y . Potom $\beta_1 = \beta_2$.

Každý úplný zväz je zväzom a každý zväz je posetom. Štruktúra úplného zväzu v spojení s Galoisovou konexiou sú základnými znalosťami, o ktoré sa opiera formálna konceptová analýza.

Nech (α, β) je Galoisova konexia medzi úplnými zväzmi X a Y .

Symbolom $L_{\alpha, \beta}$ budeme označovať množinu

$$L_{\alpha, \beta} = \{(x, y) \in X \times Y : \alpha(x) = y, \beta(y) = x\}$$

Symbolom $\leq_{\alpha, \beta}$ budeme označovať reláciu na množine $L_{\alpha, \beta}$, predpísanú vzťahom

$$(x_1, y_1) \leq_{\alpha, \beta} (x_2, y_2) \Leftrightarrow x_1 \leq_X x_2 \Leftrightarrow y_1 \geq_Y y_2$$

Veta 1.1. Nech (α, β) je Galoisova konexia medzi úplnými zväzmi X a Y . Potom nosič $L_{\alpha, \beta}$ s čiastočným usporiadaním $\leq_{\alpha, \beta}$ tvorí úplný zväz, v ktorom pre každú podmnožinu $\{(x_i, y_i) : i \in I\} \subseteq L_{\alpha, \beta}$ platí

$$\bigwedge_{i \in I} (x_i, y_i) = \left(\bigwedge_{i \in I} x_i, \alpha \left(\beta \left(\bigvee_{i \in I} y_i \right) \right) \right)$$

$$\bigvee_{i \in I} (x_i, y_i) = \left(\beta \left(\alpha \left(\bigvee_{i \in I} x_i \right) \right), \bigwedge_{i \in I} y_i \right)$$

1.2 Klasická formálna konceptová analýza

Dáta, ktoré zvyčajne ukladáme do databáz sú peniazmi informatiky. Tým získame istotu, že ich nestratíme a máme možnosť ich kedykoľvek použiť. Analógia úrokov existuje tiež, t.j. v uložených dátach môžu existovať isté súvislosti, ktoré na prvý pohľad počas vkladania do databázy nepostrehneme. Niektoré záznamy sú si podobné, ba dokonca rovnaké, ďalšie tvoria prirodzené skupiny alebo sa nám podarí nájsť funkčné závislosti medzi hodnotami niektorých ich atribútov. Metódy hľadania takýchto súvislostí sa jednotne nazývajú **data-mining** (po slovensky „dolovanie v dátach“).

Formálna konceptová analýza je jednou z data-miningových metód, ktorej základné piliere postavili nemeckí matematici Bernhard Ganter a Rudolf Wille prácou Formal Concept Analysis a stala sa veľmi užitočným nástrojom pre charakterizovanie dát s objektovo-atribútovým charakterom. Nimi predkladaná teória je založená na algebraickej teórii úplných zväzov. Predmetom formálnej konceptovej analýzy je databázová tabuľka, ktorá spĺňa podmienky objektovo-atribútového modelu. Objektovo-atribútový model (viď Tab. 1) je tabuľka, ktorej riadky sú jednotlivé databázové položky (objekty) a stĺpce sú vlastnosti (atribúty) týchto položiek. Hodnota v konkrétnom riadku a stĺpci tabuľky zodpovedá príslušnosti daného objektu k atribútu.

Tab. 1 Príklad objektovo-atribútového modelu

	Mat	Dej	Bio	Ang
Patrik	×			×
Ondrej	×			×
Ivana	×		×	×
Mária	×			×
Peter			×	×

V uvedenom objektovo-atribútovom modeli (Tab. 1) sú objektmi študenti Patrik, Ondrej, Ivana, Mária a Peter a atribútmi predmety matematika, dejepis, biológia a angličtina. Hodnota v konkrétnej bunke znamená, či daný študent zodpovedajúci predmet absolvoval alebo nie. Uvedené hodnoty daných študentov istým spôsobom charakterizujú, tí však majú aj iné vlastnosti, o ktorých sa z tabuľky nedozvieme. Naším výberom vlastností, ktoré budeme skúmať sme charakteristiky študentov zúžili. Sme presvedčení, že nami vybrané vlastnosti sú z daného uhla pohľadu (**kontextu**) relevantné a vynechaním nepodstatných detailov je výsledná charakteristika postačujúca v tomto zmysle. Tým pádom formálna konceptová analýza zanedbáva isté aspekty popisovanej reality, čo je základom pre abstraktné myslenie. Formálna konceptová analýza je veľmi užitočným nástrojom pre hľadanie skrytých súvislostí medzi objektmi a atribútmi vstupnej tabuľky. Ako napovedá jej názov, kľúčovým pojmom je **koncept**, t.j. význačná skupina, ktorá vznikla zatriedovaním objektov a atribútov do význačných skupín. Z n objektov môžeme vytvoriť 2^n rôznych skupín, avšak nie každá z nich je zmysluplná. V určitom zmysle uzavreté skupiny objektov vytvárajú koncepty. Každý koncept vzniká výraznou podobnosťou svojich prvkov a ich výraznou odlišnosťou od zvyšku objektov a atribútov. Táto myšlienka je prítomná v spôsoboch definovania množiny. Buď ju definujeme extenzionálne (vymenovaním prvkov) alebo intenzionálne (vymenovaním požadovaných vlastností, ktoré majú prvky spĺňať). Extenzionálna definícia množiny predstavuje šírku informácií a intenzionálna definícia množiny hĺbku informácií, ktoré získavame. V ďalšom texte tieto poznatky formálne zadefinujeme.

Definícia 1.10. Nech X a Y sú neprázdne množiny a nech I je podmnožinou $X \times Y$. Potom trojicu $\langle X, Y, I \rangle$ nazývame **formálny kontext** (alebo skrátene **kontext**). Prvky množiny Y nazývame **atribúty**, prvky množiny X nazývame **objekty** a I nazývame **incidenčnou reláciou**.

Objektovo-atribútový model v Tab. 1, kde $X = \{\text{Patrik, Ondrej, Ivana, Mária, Peter}\}$ sú študenti, ktorí buď absolvovali alebo neabsolvovali predmety $Y = \{\text{mat, dej, bio, ang}\}$ je príkladom formálneho kontextu. Incidenčná relácia I je určená týmto spôsobom:

1. $\langle x, y \rangle \in I$ práve vtedy, ak študent x absolvoval predmet y
2. $\langle x, y \rangle \notin I$ práve vtedy, ak študent x neabsolvoval predmet y

Vzťahy medzi študentmi a predmetmi sú prehľadne zapísané v kontexte (Tab. 1). Absolvovanie predmetu študentom označujeme znakom $\times$. Na kontexte Ganter s Willem definujú symetrické zobrazenia $\uparrow$ a $\downarrow$:

Definícia 1.11. Definujme zobrazenia $\uparrow: \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ a $\downarrow: \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$, ktoré prislúchajú kontextu $\langle X, Y, I \rangle$ nasledovným spôsobom ($\mathcal{P}(X)$ je označenie pre systém všetkých podmnožín množiny X):

1. Pre $A \subseteq X$, $A^\uparrow = \{y \in Y: (\forall x \in A)\langle x, y \rangle \in R\}$.
2. Pre $B \subseteq Y$, $B^\downarrow = \{x \in X: (\forall y \in B)\langle x, y \rangle \in R\}$.

Zobrazenie $\uparrow$ danej množine objektov A priradzuje maximálnu množinu atribútov splnených všetkými objektmi z A . Naopak, zobrazenie $\downarrow$ priradzuje množine atribútov B maximálnu množinu objektov spĺňajúcich všetky atribúty z B . Použitie jednotlivých operátorov je znázornené v Tab. 2 a Tab. 3. Napríklad, pre náš kontext platí $X^\uparrow = \{\text{ang}\}$ (Tab. 2) a $\{\text{mat, ang}\}^\downarrow = \{\text{Patrik, Ondrej, Ivana, Mária}\}$ (Tab. 3).

Tab. 2 Príklad použitia zobrazenia $\uparrow$

	Mat	Dej	Bio	Ang
Patrik	×			×
Ondrej	×			×
Ivana	×		×	×
Mária	×			×
Peter			×	×

Tab. 3 Príklad použitia zobrazenia $\downarrow$

	Mat	Dej	Bio	Ang
Patrik	×			×
Ondrej	×			×
Ivana	×		×	×
Mária	×			×
Peter			×	×

$X^\uparrow = \{\text{ang}\}$ je pre nás veľmi zaujímavým príkladom, pretože taktiež platí $\{\text{ang}\}^\downarrow = X$. Je potrebné si uvedomiť že tento symetrický vzťah medzi množinou všetkých študentov X a množinou predmetov $\{\text{ang}\}$ nám hovorí, že všetci študenti absolvovali angličtinu. Inak povedané našu množinu popisujeme extenzionálne, vymenovaním študentov (objektov) alebo intenzionálne, vymenovaním vlastností, ktoré jej prvky spĺňajú, vymenovaním predmetov (atribútov). Množina všetkých študentov X spolu s množinou predmetov $\{\text{ang}\}$ tvoria **koncept**. Takýto koncept môžete spokojne nazvať napríklad angličtinári. V tomto momente je čitateľovi jasné kam naša snaha bude smerovať. Chceme nájsť všetky koncepty, ktoré sa v kontexte nachádzajú. Pred tým si pojem koncept formálne zavedme v definícii nižšie.

Definícia 1.12. Nech $\langle X, Y, I \rangle$ je formálny kontext vybavený zobrazeniami $\uparrow$ a $\downarrow$. Ak podmnožiny $A \subseteq X$ a $B \subseteq Y$ sú také, že platí $A^\uparrow = B$ a súčasne $B^\downarrow = A$, tak dvojicu

$\langle A, B \rangle$ nazývame **formálny koncept** kontextu $\langle X, Y, I \rangle$. Množinu A nazývame **extentom** a množinu B nazývame **intentom** konceptu $\langle A, B \rangle$.

Je zrejmé, že $\langle A, B \rangle$ je konceptom práve vtedy, keď $A = A^{\uparrow\downarrow}$ a $B = A^{\uparrow}$. Taktiež môžeme povedať, že koncept je dvojica uzavretých množín vzhľadom na Galoisovu konexiu a uzavreté uzáverovým operátorom danej Galoisovej konexie.

V našom kontexte sa nachádza 5 konceptov:

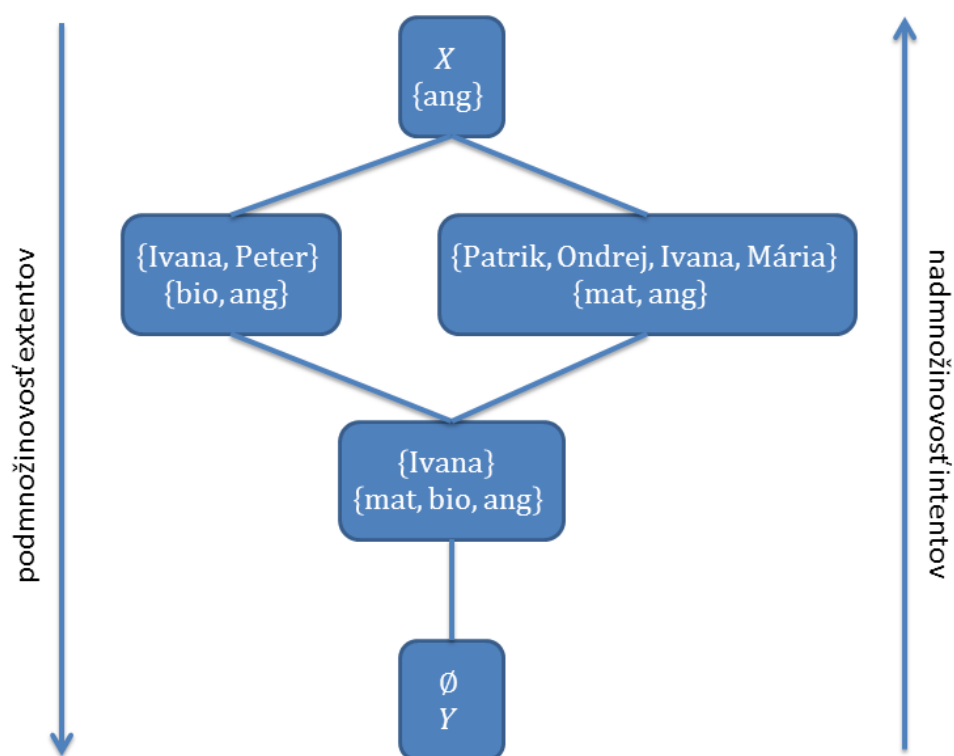
1. $\langle \{\text{Ivana}, \text{Peter}\}, \{\text{bio}, \text{ang}\} \rangle$
2. $\langle \{\text{Patrik}, \text{Ondrej}, \text{Ivana}, \text{Mária}\}, \{\text{mat}, \text{ang}\} \rangle$
3. $\langle X, \{\text{ang}\} \rangle$
4. $\langle \emptyset, Y \rangle$
5. $\langle \{\text{Ivana}\}, \{\text{mat}, \text{bio}, \text{ang}\} \rangle$

Usporiadáním všetkých konceptov vzhľadom na podmnožinovosť extentov (alebo vzhľadom na nadmnožinovosť intentov) získame štruktúru znázornenú na Obr. 1. Táto štruktúra tvorí úplný zväz podľa vety 1.1. Systémy množín $\mathcal{P}(X)$ a $\mathcal{P}(Y)$ usporiadané inklúziou sú úplné zväzy. Dvojica zobrazení $(\uparrow, \downarrow)$ je Galoisovou konexiou medzi úplnými zväzmi $(\mathcal{P}(X), \subseteq)$ a $(\mathcal{P}(Y), \subseteq)$. S ohľadom na definíciu 1.9 môžeme uviesť nasledujúcu lemu. Každá z vymenovaných vlastností objasňuje výhody Galoisovej konexie.

Lema 1.5. Nech $A, A_1, A_2 \subseteq X$ a $B, B_1, B_2 \subseteq Y$. Potom platí:

1. Ak $A_1 \subseteq A_2$, tak $A_1^{\uparrow} \supseteq A_2^{\uparrow}$.
2. Ak $B_1 \subseteq B_2$, tak $B_1^{\downarrow} \supseteq B_2^{\downarrow}$.
3. $A \subseteq (A^{\uparrow})^{\downarrow}$.
4. $B \subseteq (B^{\downarrow})^{\uparrow}$.

Vlastnosť 1 vyjadruje prirodzenú vec, že čím viac objektov, tým menej spoločných vlastností. Vlastnosť 2 hovorí, že čím viac podmienok na objekty položíme, tým menej ich bude všetkým vyhovovať. Vlastnosti 3 a 4 zdôrazňujú, že pri oboch zloženiach zobrazení sa žiaden objekt, resp. atribút nestratí.



Obr. 1 Hasseho čiarový diagram pre úplný zväz konceptov kontextu v Tab. 1

Definícia 1.13. Nech $\langle X, Y, I \rangle$ je formálny kontext vybavený dvojicou zobrazení $(\uparrow, \downarrow)$, ktorá je jednou z Galoisových konexií medzi úplnými zväzmi $(\mathcal{P}(X), \subseteq)$ a $(\mathcal{P}(Y), \subseteq)$. Potom **formálnym konceptovým zväzom** nazývame úplný zväz $CL(X, Y, I) = (L_{\uparrow, \downarrow}, \leq_{\uparrow, \downarrow})$.

1.3 Fuzzy formálna konceptová analýza

Klasický prístup Gantera a Willeho pracuje s dvojhodnotovým kontextom, prirodzene teda vzniká otázka, ako postupovať v prípade, keď príslušný objektovo-atribútový model pripúšťa viacero hodnôt. V príklade z predchádzajúcej podkapitoly sme mali k dispozícii 2 možnosti – absolvoval alebo neabsolvoval. Napríklad ak chceme modelovať dosiahnuté hodnotenia študentov v daných predmetoch, tak predošlý prístup nepostačuje našim potrebám. Zvolili sme si všeobecnejší **fuzzy** prístup, ktorý bol vytvorený tímom okolo prof. Bělohlávka v článkoch [7], [8] a [2]. Tento prístup si uvedieme v nasledujúcich riadkoch, pred tým si ale uvedme všetky potrebné znalosti.

1.3.1 Fuzzy množina

Definícia 1.14. Nech L je úplný reziduovaný zväz. L -množina A v priestore prvkov X je zobrazenie, ktoré každému prvku $x \in X$ priradí pravdivostnú hodnotu $A(x) \in L$.

Pravdivostnú hodnotu $A(x)$ interpretujeme ako stupeň príslušnosti prvku x k L -množine A . Množinu všetkých L -množín v priestore X označujeme L^X .

Definícia 1.15. Uvažujme L -množiny $A, B \in L^X$. Množina A je **podmnožinou** B , zapisujeme $A \subseteq B$, ak pre každý prvok $x \in X$ platí $A(x) \leq B(x)$.

1.3.2 Lukasiewiczova logika

V našej práci používame Lukasiewiczovu logiku, ktorá patrí k t-norm logikám a subštruktúrnym logikám. V tejto logike je operácia t-norm $\otimes$ (silná Lukasiewiczova konjunkcia) definovaná nasledovne:

$$k \otimes l = \max(0, k + l - 1)$$

Na definovanie zobrazení, ktoré prislúchajú kontextu, a teda aj konceptov používame operáciu reziduum (vyvolaná t-normom) $\rightarrow$ (Lukasiewiczova implikácia), ktorá je definovaná nasledovne:

$$k \rightarrow l = \min(1, 1 - k + l)$$

Majme trojprvkovú Lukasiewiczovu logiku $L = \{0, 0.5, 1\}$, $\min, \max, \otimes, \rightarrow, 0, 1$. Pravdivostné stupne Lukasiewiczovej silnej konjunkcie a implikácie sú uvedené nižšie v tabuľkách Tab. 4 a Tab. 5.

Tab. 4 Lukasiewiczova silná konjunkcia $\otimes$

$\otimes$	1	0.5	0
1	1	0.5	0
0.5	0.5	0	0
0	0	0	0

Tab. 5 Lukasiewiczova implikácia $\rightarrow$

$\rightarrow$	1	0.5	0
1	1	0.5	0
0.5	1	1	0.5
0	1	1	1

1.3.3 Fuzzy uzáverový operátor

Fuzzy uzáverový operátor na priestore X je zobrazenie $C: L^X \rightarrow L^X$, pričom musia platiť nasledovné pravidlá:

1. $A \subseteq C(A)$
2. Ak $A \subseteq B$, tak $C(A) \subseteq C(B)$
3. $C(A) = C(C(A))$

pre všetky $A, B \in L^X$.

Ak $A = C(A)$, tak A nazývame fixpoint uzáverového operátora C . Množinu všetkých fixpointov uzáverového operátora C označíme $\text{fix}(C)$, t.j. $\text{fix}(C) = \{A \in L^X | A = C(A)\}$.

1.3.4 Formálna konceptová analýza dát s fuzzy atribútmi

Klasická podoba formálnej konceptovej analýzy pracuje s dvoma hodnotami - áno, resp. nie. Existuje však mnoho situácií, keď tieto dve hodnoty nedokážu skúmané vzťahy plne popísať. V takom prípade potrebujeme na ich charakterizáciu viac hodnôt, tým pádom použijeme viachodnotovú verziu. Výborným kandidátom pre zhostenie sa tejto funkcie je fuzzy formálna konceptová analýza, ktorá pracuje s dátami obsahujúcimi fuzzy atribúty. Takéto dáta znázorňujeme tabuľkou a formálne uvádzame v nasledujúcej definícii (informácie boli čerpané z článkov [3] a [7]):

Definícia 1.16. Nech L je úplný reziduovaný zväz. **L -kontext** je usporiadaná trojica $\langle X, Y, I \rangle$ obsahujúca priestor objektov X , priestor atribútov Y a L -fuzzy binárnu reláciu I , čiže zobrazenie $I: X \times Y \rightarrow L$. Zobrazenie I alternatívne rozumieme ako L -fuzzy podmnožinu súčinu $X \times Y$.

Príklad L -kontextu uvádzame v Tab. 6. Hodnoty $\{1,2,3,4,5\}$ korešpondujú s množinou pravdivostných hodnôt $L = \{1,0.75,0.5,0.25,0\}$.

Tab. 6 Výsledky študentov v okruhoch predmetov

	Mat	Dej	Bio	Ang
Patrik	1	2	2	1
Ondrej	1	3	3	1
Ivana	1	1	1	1

Pre každý L -kontext $\langle X, Y, I \rangle$ definujeme zobrazenia $\uparrow: L^X \rightarrow L^Y$ a $\downarrow: L^Y \rightarrow L^X$, ktoré sú rozšírením zobrazení (derivačných operátorov) v dvojhodnotovom prípade:

Pre všetky L -množiny objektov $A \in L^X$ určujeme L -množiny atribútov $A^\uparrow \in L^Y$ nasledovne:

$$A^\uparrow(y) = \bigwedge_{x \in X} (A(x) \rightarrow I(x, y))$$

Pre všetky L -množiny atribútov $B \in L^Y$ určujeme L -množiny objektov $B^\downarrow \in L^X$ nasledovne:

$$B^\downarrow(x) = \bigwedge_{y \in Y} (B(y) \rightarrow I(x, y))$$

$A^\uparrow(y)$ je stupeň akým je atribút y zdieľaný všetkými objektmi z L -množiny A a analogicky $B^\downarrow(x)$ je stupeň, akým objekt x vlastní všetky atribúty z L -množiny B . Teda zobrazenie $\uparrow$ priradzuje L -množine objektov L -množinu atribútov, ktoré patria objektom vstupnej množiny a zobrazenie $\downarrow$ priradzuje L -množine atribútov L -množinu objektov, ktoré zdieľajú atribúty vstupnej množiny. Dvojica vyššie definovaných zobrazení tvorí Galoisovu konexiu medzi úplnými zväzmi $\langle L^X, \subseteq \rangle$ a $\langle L^Y, \subseteq \rangle$, t.j. zobrazenia $\uparrow: L^X \rightarrow L^Y$ a $\downarrow: L^Y \rightarrow L^X$ sú antitónne. Navyše, pre každú L -množinu $A \in L^X$ a $B \in L^Y$ platí $A \subseteq A^{\uparrow\downarrow}$ a $B \subseteq B^{\downarrow\uparrow}$ (t.j. $A(x) \leq A^{\uparrow\downarrow}(x)$ a $B(y) \leq B^{\downarrow\uparrow}(y)$, pre objekt x a atribút y).

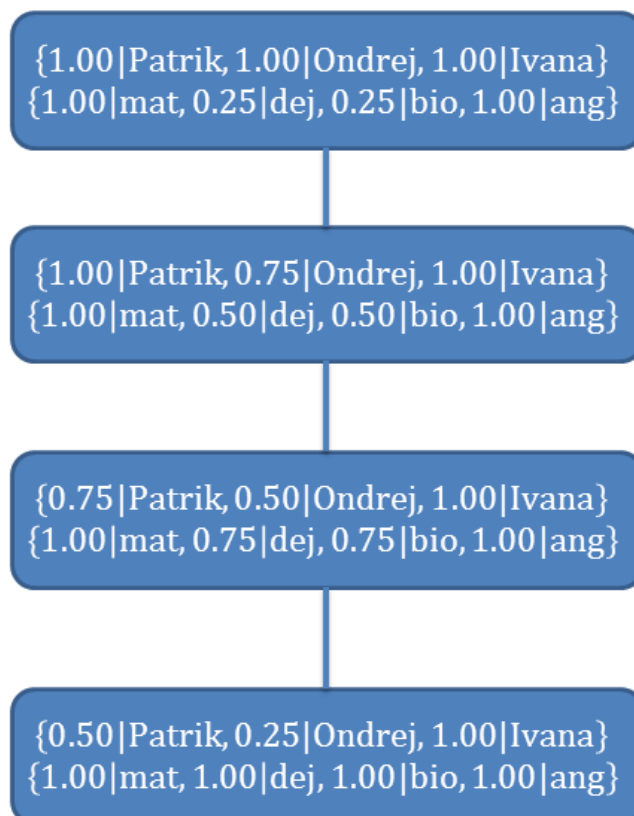
Definícia 1.17. Majme L -kontext $\langle X, Y, I \rangle$. L -množinu objektov $A \in L^X$ nazývame **uzavretou** práve vtedy, keď $A = A^{\uparrow\downarrow}$. Analogicky L -množinu atribútov $B \in L^Y$ nazývame **uzavretou** práve vtedy, keď $B = B^{\downarrow\uparrow}$.

Každú usporiadanú dvojicu $\langle A, B \rangle \in L^X \times L^Y$, pre ktorú platí $A^\uparrow = B$ a $B^\downarrow = A$ nazývame **formálny fuzzy koncept**. Množinu A nazývame extent a množinu B nazývame intent. Množinu všetkých konceptov označujeme $CL(X, Y, I)$ a nazývame **formálny fuzzy konceptový zväz**. Pre zjednodušenie množinu všetkých intentov označujeme $Int(X, Y, I)$ a množinu všetkých extentov označujeme $Ext(X, Y, I)$.

$$Int(X, Y, I) = \{B \in L^Y \mid \langle A, B \rangle \in CL(X, Y, I) \exists A \in L^X\}$$

$$Ext(X, Y, I) = \{A \in L^X \mid \langle A, B \rangle \in CL(X, Y, I) \exists B \in L^Y\}$$

Hierarchiu konceptov vo fuzzy konceptovom zväze, ktorá stvárňuje dobre známy subkoncept-superkonceptový vzťah modelujeme čiastočným usporiadaním $\leq$ nasledovne: fuzzy koncept $\langle A_1, B_1 \rangle$ je menší ako fuzzy koncept $\langle A_2, B_2 \rangle$, zapisujeme $\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle$ práve vtedy, keď $A_1 \subseteq A_2$ ($B_2 \subseteq B_1$). Množina všetkých nájdených konceptov pre kontext v Tab. 6 je znázornená **Hasseho** čiarovým diagramom na Obr. 2. Na Obr. 2 pozorujeme, že úplný zväz intentov je **duálne izomorfný** s úplným zväzom extentov. To znamená, že usporiadanie intentov je opačné ako usporiadanie extentov.

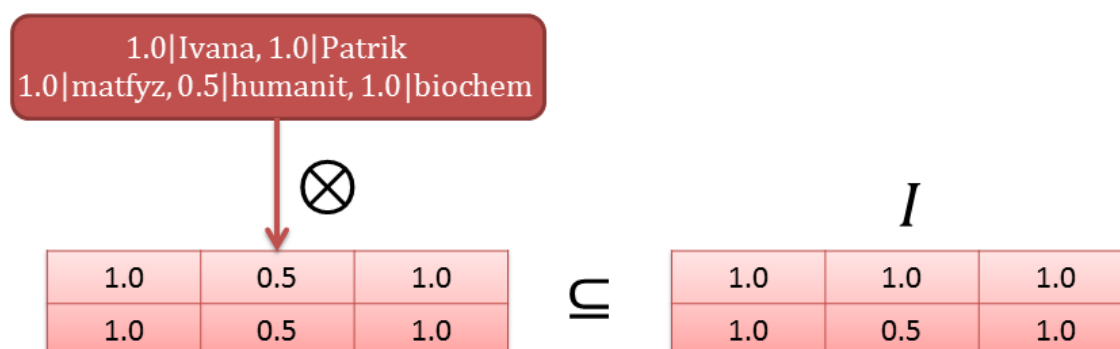


Obr. 2 Fuzzy konceptový zväz fuzzy kontextu Tab. 6

Koncepty možno alternatívne definovať ako **maximálne obdĺžniky** obsiahnuté v relácii I . Pre usporiadaný pár $\langle A, B \rangle \in L^X \times L^Y$ definujeme fuzzy reláciu $A \otimes B \in L^{X \times Y}$ vzťahom:

$(A \otimes B)(x, y) = A(x) \otimes B(y)$, t.j. $A \otimes B$ je karteziánsky súčin L -množín A a B .

Na Obr. 3 vidíme vybudovanie maximálneho obdĺžnika z konceptu pomocou karteziánskeho súčinu.



Obr. 3 Fuzzy koncept ako maximálny obdĺžnik v relácii I

Veta 1.2 je základnou vetou pre konceptové zväzy a reflektuje vlastnosti Galoisových konexií a úplných zväzov na množinu konceptov.

Veta 1.2. Poset $\langle \text{CL}(X, Y, I), \leq \rangle$ je úplný zväz, pričom platí

$$\bigwedge_{j \in J} \langle A_j, B_j \rangle = \langle \bigwedge_{j \in J} A_j, \left(\bigwedge_{j \in J} A_j \right)^\uparrow \rangle$$

a

$$\bigvee_{j \in J} \langle A_j, B_j \rangle = \langle \left(\bigwedge_{j \in J} B_j \right)^\downarrow, \bigwedge_{j \in J} B_j \rangle$$

Veta vraví, že množina všetkých konceptov tvorí úplný zväz vzhľadom na množinovú inklúziu a infimum a suprénum každej podmnožiny prvkov úplného zväzu patrí do zväzu.

2 Ako hľadať koncepty v kontexte

V tejto kapitole si predstavíme rýchly „**bottom-up**“ algoritmus hľadania všetkých uzavretých množín (fixpointov) fuzzy uzáverového operátora spolu s ich čiastočným usporiadaním, ktorý sme čerpali od autorov článku [2]. Fuzzy uzáverové operátory sa používajú vo viacerých oblastiach, kde sa aplikuje fuzzy logika, vrátane formálnej konceptovej analýzy a spolu s fuzzy Galoisovými konexiami tvoria matematický základ formálnej konceptovej analýzy. Sú to zobrazenia, ktoré L -množinám priradujú L -množiny. Ústredným cieľom fuzzy formálnej konceptovej analýzy je objavenie hierarchickej štruktúry (tzv. fuzzy konceptový zväz), ktorá pozostáva z **klastrov** (fuzzy formálnych konceptov) asociovaných so vstupnými dátami, ktoré majú fuzzy atribúty a nájdenie **dátových závislostí** (tzv. atribútových implikácií). Informácie o subkonceptoch a superkonceptoch majú rôznorodé využitie u užívateľov, napríklad, tvorba Hasse diagramov konceptových zväzov. Keďže formálne koncepty sú práve fixpointy konkrétneho uzáverového operátora, ktorý je asociovaný so vstupnými dátami, tak problém hľadania všetkých konceptov sa dá elegantne redukovať na problém nájdenia všetkých uzavretých množín tohto uzáverového operátora. Touto znalosťou sa zaručuje rýchlosť predstaveného algoritmu oproti brute-force algoritmu.

2.1 Problém: nájsť zväz všetkých fixpointov

2.1.1 Predbežné úvahy

Je potrebné si najprv uvedomiť, že nie je zrejmé ako efektívne nájsť všetky fixpointy fuzzy uzáverového operátora C . Definícia nám ponúka iba brute-force prístup: Vygenerujte všetky L -množiny $A \in L^X$ a vráťte tie, pre ktoré platí $A = C(A)$. Je zrejmé, že takýto prístup má exponenciálnu časovú zložitosť a je preto nepoužiteľný.

2.1.2 Spojitosť medzi uzáverovými operátormi a konceptovými zväzmi

Je známe, že pre formálny kontext $\langle X, Y, I \rangle$ zložené zobrazenie $\uparrow\downarrow: L^X \rightarrow L^X$ je uzáverový operátor na priestore X a analogicky zložené zobrazenie $\downarrow\uparrow: L^Y \rightarrow L^Y$ je uzáverový operátor na priestore Y . Fixpointy uzáverového operátora $\uparrow\downarrow$ sú práve extenty kontextu $\langle X, Y, I \rangle$ a analogicky fixpointy uzáverového operátora $\downarrow\uparrow$ sú práve intenty kontextu $\langle X, Y, I \rangle$. Skrátene napísané $\text{fix}(\uparrow\downarrow) = \text{Ext}(X, Y, I)$ a $\text{fix}(\downarrow\uparrow) = \text{Int}(X, Y, I)$.

Každý formálny koncept je daný extentom A ($B = A^\uparrow$) alebo intentom B ($A = B^\downarrow$). Preto $CL(X, Y, I) = \{\langle B^\downarrow, B \rangle \mid B \in \text{Int}(X, Y, I) = \langle B^\downarrow, B \rangle \mid B \in \text{fix}(\downarrow\uparrow)\}$. To znamená, že problém hľadania všetkých konceptov sa **redukuje** buď na problém hľadania všetkých intentov, t.j. všetkých fixpointov uzáverového operátora $\downarrow\uparrow$, alebo na problém hľadania všetkých extentov, t.j. všetkých fixpointov uzáverového operátora $\uparrow\downarrow$.

2.1.3 Cieľ algoritmu

Je daný fuzzy uzáverový operátor C na priestore prvkov Y

1. Nájdite všetky fixpointy uzáverového operátora C
2. Pre každý fixpoint $A \in \text{fix}(C)$ nájdite množiny jeho priamych horných a dolných susedov s ohľadom na $\subseteq$ (inklúzia na L -množinách)

Teda naším cieľom je nájsť všetky fixpointy uzáverového operátora C s prislúchajúcim čiastočným usporiadaním, alebo inak povedané tento problém je možné interpretovať ako problém hľadania konceptových zväzov. V podstate sa zaoberáme hľadaním konceptového zväzu $CL(X, Y, I)$ spolu s subkoncept-superkonceptovým usporiadaním.

Ani definícia fuzzy uzáverového operátora, ani navrhnutý algoritmus nebudú využívať reziduum, či multiplikáciu na množine L (konečný reťazec pravdivostných hodnôt). Algoritmus pracuje iba s reťazcom pravdivostných hodnôt a využíva iba jeho usporiadanie. Keďže pracujeme s formálnou konceptovou analýzou, reziduum bolo použité iba na definíciu uzáverových operátorov.

2.2 Horní susedia a ich generátory

Nech $L = \{a_1, \dots, a_k\}$ (množina pravdivostných hodnôt), pričom $0 = a_1 < \dots < a_k = 1$

Definícia 2.1. $D \in \text{fix}(C)$ nazývame **horným susedom** $B \in \text{fix}(C)$ s ohľadom na C , ak platí nasledovné:

1. $B \subset D$.
2. Neexistuje $D' \in \text{fix}(C)$, že $B \subset D' \subset D$.

Analogicky možno definovať dolných susedov.

Ak C je $\downarrow\uparrow$, tak horní susedia intentu B s ohľadom na uzáverový operátor $\downarrow\uparrow$ sú intenty priamych subkonceptov konceptu $\langle B^\downarrow, B \rangle$, pretože väčšie intenty prislúchajú menším konceptom (prihliadajúc na príslušné usporiadania). Analogicky, ak C je $\uparrow\downarrow$, tak horní susedia extentu A sú extenty priamych superkonceptov konceptu $\langle A, A^\uparrow \rangle$.

Definícia 2.2. Nech $[y]_B^C$ znamená $C(B \cup \{B(y)^+ / y\})$, pre každý fixpoint $B \in L^Y$ a prvok $y \in Y$, taký že $B(y) < 1$. Ak fixpoint $[y]_B^C$ je horným susedom B , tak $[y]_B^C$ nazývame **horný sused generovaný prvkom y** .

V predošlej definícii sme predstavili pojem generátor horných susedov.

$C(B \cup \{B(y)^+ / y\})$ znamená uzáver novej L -množiny, ktorá vznikla nahradením existujúcej pravdivostnej hodnoty ďalšou v poradí v rámci reťazca L . Týmto ešte raz pripomíname, že algoritmus využíva iba čiastočné usporiadanie reťazca L . Myšlienka generovania horných susedov je odzrkadlená vo funkcii **NEIGHBORS**, ktorá akceptuje ako vstupy uzáverový operátor $C: L^Y \rightarrow L^Y$ a fixpoint $B \in \text{fix}(C)$ a vráti množinu všetkých horných susedov fixpointu na vstupe.

Function NEIGHBORS(B, C)	
1	$U := \emptyset;$
2	$Min := \{y \in Y \mid B(y) < 1\};$
3	foreach $y \in Y$ such that $B(y) < 1$ do
4	$D := [y]_B^C;$
5	$Increased := \{z \in Y \mid z \neq y \text{ and } B(z) < D(z)\};$
6	if $Min \cap Increased = \emptyset$ then
7	add D to U
8	else
9	remove y from Min
10	end
11	end
12	return U

Algoritmus používa premenné U a Min . U je množina všetkých horných susedov, ktorá je na začiatku prázdna a Min je množina prvkov z priestoru Y , ktorú vnímame ako množinu možných generátorov horných susedov fixpointu B . Množina Min je na začiatku naplnená prvkami, ktorých pravdivostná hodnota je menšia ako 1, t.j. $Min := \{y \in Y | B(y) < 1\}$. Počas behu algoritmu z množiny Min sú odstránené tie prvky y , ktoré nie sú generátory horných susedov fixpointu B . Na konci algoritmu sa v množine Min nachádzajú iba tie prvky, ktoré sú generátormi horných susedov fixpointu B . Ako je možné vidieť z riadkov 4, 7, 9 vo funkcii **NEIGHBORS**, prvok y ostane v množine Min práve vtedy, ak fixpoint $D = [y]_B^C$ je pridaný do množiny U . Ďalej *Increased* je množina prvkov priestoru Y , ktoré patria fixpointu $D = [y]_B^C$ a ich pravdivostné hodnoty sú ostro väčšie ako pravdivostné hodnoty fixpointu B (viď riadok 5 funkcie **NEIGHBORS**). Taktiež vynechávame práve spracovávaný prvok y . Test na riadku 6 funkcie **NEIGHBORS** je úspešný práve vtedy, ak $D = [y]_B^C$ je horným susedom fixpointu B , pričom prvok y je generátorom fixpointu D . Preto prvok y môžeme nechať v množine Min a fixpoint D je pridaný do výslednej množiny U .

2.3 Algoritmus pre hľadanie všetkých fixpointov

Algoritmus pre hľadanie všetkých fixpointov začína s najmenším fixpointom uzáverového operátora C , t.j. uzáver prázdnej množiny $C(\emptyset)$ a pridá ho do množiny nájdených fixpointov. Pre každý novopridaný fixpoint zistíme jeho horných susedov použitím funkcie **NEIGHBORS** a potom aktualizujeme informácie o dolných susedoch (D je horným susedom B práve vtedy, keď B je dolným susedom D). Pre každého horného suseda, ktorý nebol nájdený v predošlých krokoch algoritmu rekurzívne opakujeme proces, kým nenarazíme na najväčší fixpoint uzáverového operátora C , t.j. plná množina Y . Algoritmus je rozdelený do dvoch funkcií: **LATTICE** a **GENERATEFROM**. Funkcia **LATTICE** akceptuje ako vstupy uzáverový operátor $C: L^Y \rightarrow L^Y$ a priestor Y a iniciuje rekurzívne generovanie fixpointov začínajúc s najmenším. Pomocná funkcia **GENERATEFROM** si plní úlohu generovania fixpointov.

Function GENERATEFROM(B)

```
1 while  $B \neq Y$  do
2    $B^* := \text{NEIGHBORS}(B, C);$ 
3    $\mathcal{N} := B^* - F;$ 
4   foreach  $D \in B^*$  do
5     add  $B$  to  $D_*$ ;
6     if  $D \in \mathcal{N}$  then
7       add  $D$  to  $F$ ;
8     end
9   end
10  foreach  $D \in \mathcal{N}$  do
11    GENERATEFROM( $D$ )
12  end
13 end
14 return
```

Function LATTICE(C, Y)

```
1  $F := \emptyset;$ 
2  $B := C(\emptyset);$ 
3 add  $B$  to  $F$ ;
4 GENERATEFROM( $B$ );
5 return  $\langle F, \{Y^* | Y \in F\}, \{Y_* | Y \in F\} \rangle$ 
```

Obe funkcie používajú premenné F , B^* , B_* . F uchováva fixpointy, ktoré boli nájdené v predošlých krokoch. B^* je množina všetkých horných susedov fixpointu B a B_* je množina všetkých dolných susedov fixpointu B . N je lokálna premenná, ktorá reprezentuje nové fixpointy, ktoré boli nájdené v konkrétnom rekurzívnom volaní funkcie **GENERATEFROM**. Funkcia **LATTICE** je korektná, pretože každý fixpoint uzáverového operátora C je spracovaný práve raz vo funkcii **GENERATEFROM**. Je to

zaistené na riadku 11 funkcie **GENERATEFROM**, kde je funkcia **GENERATEFROM** rekurzívne zavolaná iba pre tie fixpointy, ktoré sa zatiaľ nenašli. Informácie o dolných susedoch sú aktualizované správne, lebo každý fixpoint B je dolným susedom iba tých fixpointov, ktorý sú hornými susedmi B . Je evidentné, že algoritmus skončí po konečne veľa krokoch, pretože funkcia **GENERATEFROM** je volaná práve raz pre každý fixpoint.

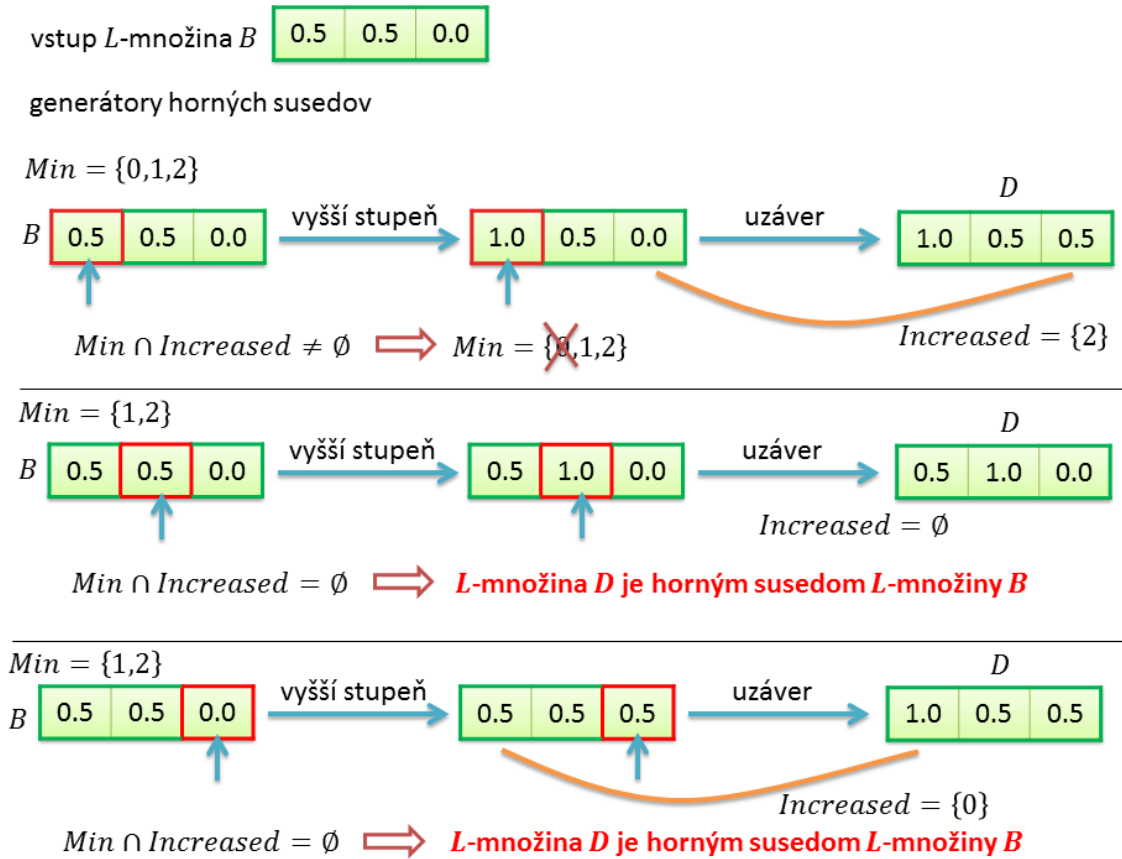
2.4 Príklad výpočtu

V tejto časti si ukážeme príklad výpočtu na názornom príklade. Pre vstup algoritmu sme zvolili kontext uvedený v Tab. 7.

Tab. 7 Ilustračný L -kontext

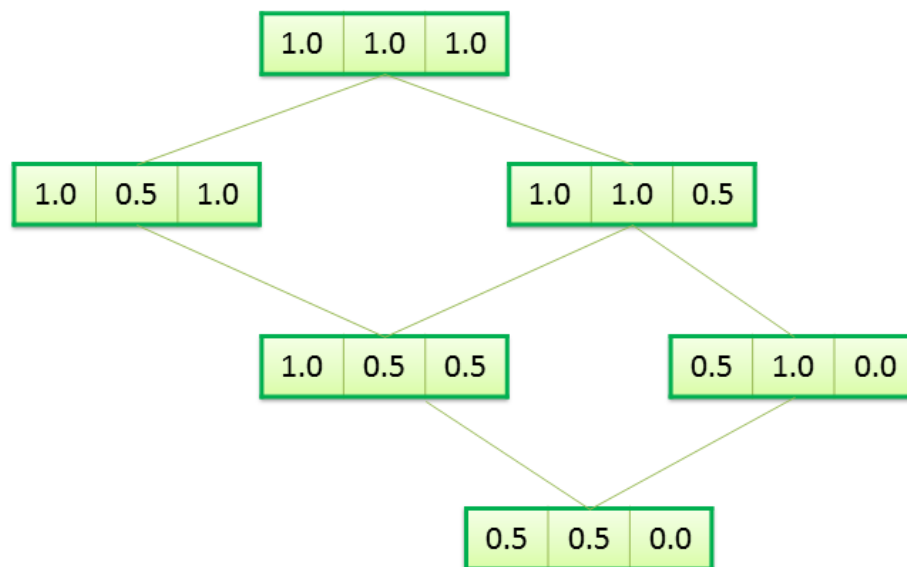
1.0	1.0	0.5	1.0
1.0	1.0	1.0	0.5
1.0	0.5	0.0	1.0

Algoritmus začína uzáverom najmenšieho fixpointu, t.j. prázdnej L -množiny a pridá ho do množiny nájdených fixpointov. Pokračujeme jednou názornou iteráciou funkcie **NEIGHBORS** na Obr. 4, ktorá nájde všetkých horných susedov ľubovoľného fixpointu. Začneme výberom iba tých prvkov, ktorých pravdivostná hodnota je menšia než 1. Faktom je, že pravdivostná hodnota 1 je najväčšou v reziduovanom zväze L , a teda je zbytočné sa ňou zaoberať. V danom fixpointe nahradíme jeden z jeho stupňov od neho vyšším a uzavrieme ho. Podľa riadku 5 pseudokódu funkcie **NEIGHBORS** zistíme všetky prvky (okrem práve spracovávaného), ktorých stupne sa zvýšili. Urobíme prienik množiny Min s nájdenou množinou a ak je prázdny, tak sme našli horného suseda vstupného fixpointu. Inak odstránime práve spracovávaný prvok z množiny Min .



Obr. 4 Pribeh funkcie NEIGHBORS pre najmenší fixpoint nášho príkladu

Výstupom funkcie **NEIGHBORS** je množina horných susedov vstupného fixpointu. V našom prípade sme pre najmenší fixpoint B našli dvoch horných susedov (viď Obr. 5). Aby sme vybudovali úplný zväz fixpointov, potrebujeme nejakú rekurzívnu funkciu, ktorá sa zastaví keď vygeneruje najväčší fixpoint, t.j. plná L -množina. Túto záležitosť rieši funkcia **GENERATEFROM**, ktorá generuje všetkých horných susedov vstupného fixpointu a vytvára medzi nimi prepojenia horný/dolný sused. Je volaná rekurzívne pre každého horného suseda vstupného fixpointu. Nakoniec sa funkcia **GENERATEFROM** zavolá pre najmenší fixpoint vo funkcii **LATTICE**, ktorá sa stará o začiatkové inicializácie premenných a vytvorenie najmenšieho fixpointu. Výstupom funkcie **LATTICE** pre L -kontext uvedený v Tab. 7 je úplný zväz fixpointov na Obr. 5.



Obr. 5 Úplný zváz fixpointov

3 Ako hľadať putá medzi dvoma kontextami

V predchádzajúcej kapitole sme ukázali ako sa hľadajú koncepty vstupného kontextu, inak povedané **významy vo vstupnom kontexte**. V tejto kapitole sa čitateľ dozvie ako hľadať **významy medzi dvoma kontextami**, inak povedané **putá**, t.j. relácie medzi priestorom objektov prvého kontextu a priestorom atribútov druhého kontextu, také že ich riadky a stĺpce sú uzavreté množiny v príslušnom kontexte. Zdôrazňujeme, že myšlienka použitá pri návrhu algoritmu bola inšpirovaná algoritmom z predchádzajúcej kapitoly a rozšírená pre hľadanie pút medzi dvoma kontextami spolu so štruktúrou. Dôvodom znovu použitia myšlienky „horných susedov“ je fakt, že množina všetkých konceptov tvorí úplný zväz vzhľadom na množinovú inklúziu a množina všetkých pút tvorí úplný zväz vzhľadom na relačné usporiadanie. V nasledujúcich riadkoch si predstavíme potrebné znalosti, ktoré sme čerpali z článku [3].

3.1 Multifunkcia, bond

Definícia 3.1. L -multifunkcia z priestoru X do priestoru Y je zobrazenie $\varphi: X \rightarrow L^Y$. Transpozícia L -multifunkcie $\varphi: X \rightarrow L^Y$ je L -multifunkcia $\varphi^T: X \rightarrow L^Y$ definovaná nasledovne: $\varphi^T(y)(x) = \varphi(x)(y)$.

Množinu všetkých L -multifunkcií z priestoru X do Y označujeme $L\text{-}Mfn(X, Y)$. Táto množina sa stane posetom, keď zdefinujeme čiastočné usporiadanie nasledovne:

$\varphi_1 \leq \varphi_2$ práve vtedy, keď $\varphi_1(x)(y) \leq \varphi_2(x)(y)$, pre všetky elementy $x \in X$ a $y \in Y$.

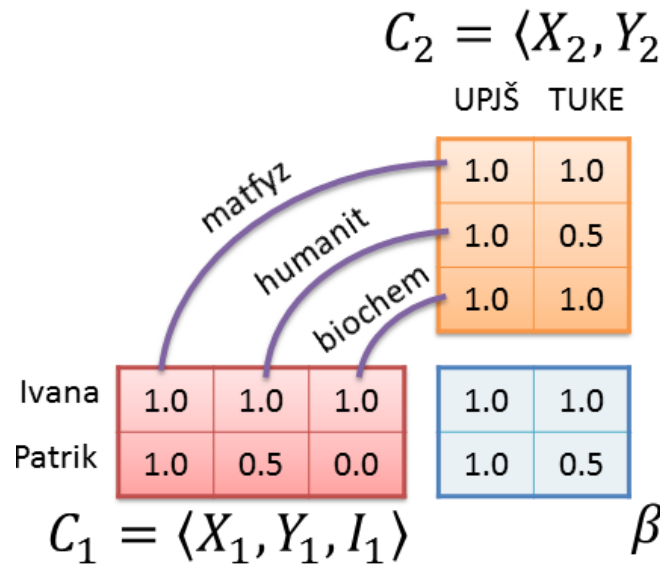
Definícia 3.2. Pre ľubovoľnú L -multifunkciu $\varphi \in L\text{-}Mfn(X, Y)$ definujeme L -fuzzy reláciu $\varphi^r \in L^{X \times Y}$ nasledovne: pre všetky páry $(x, y) \in X \times Y$ platí $\varphi^r(x, y) = \varphi(x)(y)$.

Pre ľubovoľnú L -fuzzy reláciu $r \in L^{X \times Y}$ definujeme L -multifunkciu $r^{\text{mfn}}: X \rightarrow L^Y$ nasledovne: $r^{\text{mfn}}(x)(y) = r(x, y)$.

V nasledujúcom budeme pracovať vždy s dvoma kontextami, ktoré budeme pre jednoduchosť označovať $C_1 = \langle X_1, Y_1, I_1 \rangle$ a $C_2 = \langle X_2, Y_2, I_2 \rangle$ a ich príslušné Galoisove konexie budeme označovať $\langle \uparrow_1, \downarrow_1 \rangle$ a $\langle \uparrow_2, \downarrow_2 \rangle$.

Definícia 3.3. *L*-bond medzi dvoma *L*-kontextami C_1 a C_2 je multifunkcia $\beta: X_1 \rightarrow L^{Y_2}$, ktorá spĺňa podmienku, že pre všetky $x_1 \in X_1$ a $y_2 \in Y_2$, $\beta(x_1)$ je uzavretá množina atribútov v kontexte C_2 a aj $\beta^T(y_2)$ je uzavretá množina objektov v kontexte C_1 (pričom β^T je definovaná transponovane $\beta^T: Y_2 \rightarrow L^{X_1}$).

Na Obr. 6 je znázornené puto medzi kontextami C_1 a C_2 , kde podobným uzavretým množinám študentov **spravodlivo** priradíme podobné uzavreté množiny vysokých škôl.



Obr. 6 Puto medzi dvoma kontextami

Množinu všetkých pút z kontextu C_1 do C_2 označujeme $L\text{-Bonds}(C_1, C_2)$. Množina všetkých pút spolu s hore definovaným usporiadaním *L*-multifunkcií tvorí úplný zväz o čom hovorí nasledujúca lema.

Lema 3.1. Majme *L*-kontexty C_1 a C_2 a nech $\beta_i \in L\text{-Bonds}(C_1, C_2)$. Nasledujúce operácie tvoria štruktúru úplného zväzu:

$$\left(\bigwedge_{j \in J} \beta_j \right) (x) = \bigwedge_{j \in J} \beta_j(x)$$

$$\bigvee_{j \in J} \beta_j(x) = \left(\bigvee_{j \in J} \beta_j(x) \right)^{\downarrow_2 \uparrow_2} = \left(\bigwedge_{j \in J} (\beta_j(x))^{\downarrow_2} \right)^{\uparrow_2}$$

pre $\forall x \in X_1$.

3.2 Horní susedia a ich generátory

3.2.1 Ľavý a pravý uzáver multifunkcie

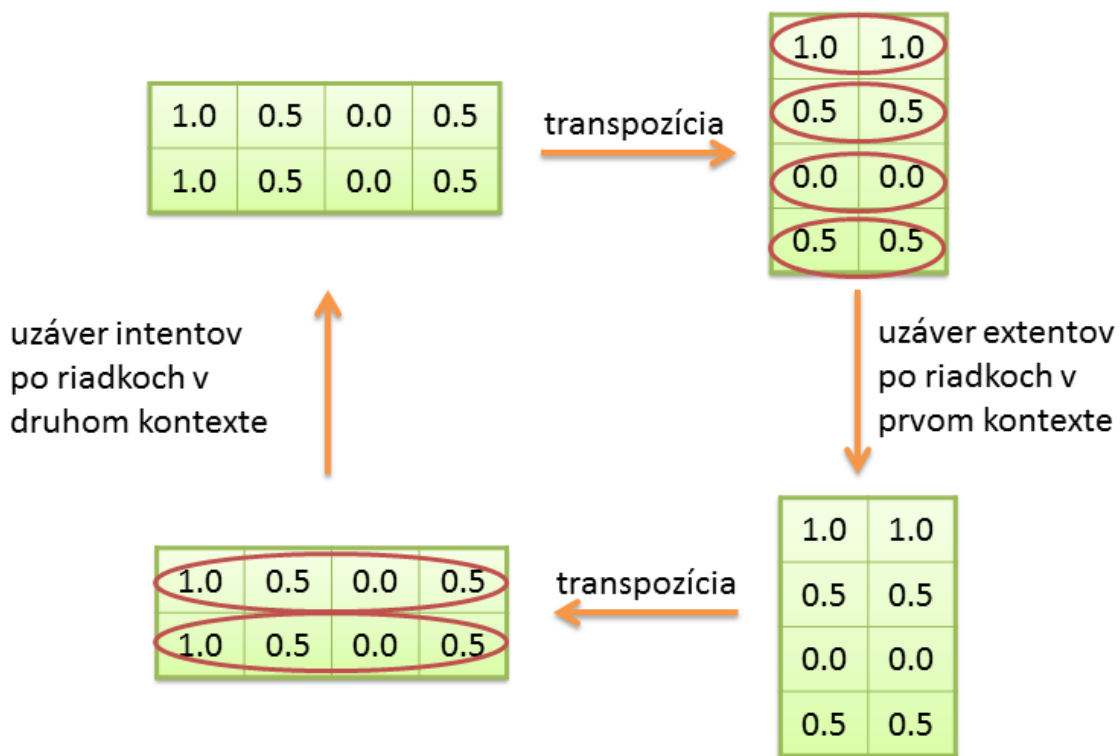
V nasledujúcich riadkoch si predstavíme ľavý a pravý uzáver multifunkcie, čo korešponduje s uzáverom L -množiny z predošlej kapitoly. Všetky detaily uvádzame v nasledujúcej definícii:

Definícia 3.4. Majme L -kontexty C_1 a C_2 . Nech $\gamma: X_1 \rightarrow L^{Y_2}$ je ľubovoľná multifunkcia medzi priestorom X_1 a Y_2 a nech $\delta: Y_2 \rightarrow L^{X_1}$ je ľubovoľná multifunkcia medzi priestorom Y_2 a X_1 . Definujme zobrazenia:

1. **Ľavý uzáver** multifunkcie γ ako $\text{leftclosure}(\gamma) = \left(\text{cl}_1 \left((\text{cl}_2(\gamma))^T \right) \right)^T$
2. **Pravý uzáver** multifunkcie δ ako $\text{rightclosure}(\delta) = \left(\text{cl}_2 \left((\text{cl}_1(\delta))^T \right) \right)^T$

kde $\text{cl}_1(\gamma)(x_1) = \uparrow_2 \downarrow_2 (\gamma(x_1))$ a $\text{cl}_2(\delta)(y_2) = \downarrow_1 \uparrow_1 (\delta(y_2))$

Pokiaľ si multifunkciu predstavujeme ako reláciu (tabuľku) medzi množinami X_1 a Y_2 (riadky predstavujú L -množiny atribútov z C_2 a stĺpce L -množiny objektov z C_1), potom uzáver cl_1 predstavuje zobrazenie, ktoré spomínanej multifunkcii priradí novú multifunkciu, ktorej stĺpce sú uzavreté pomocou zloženia Galoisových konexií prvého kontextu a podobne uzáver cl_2 predstavuje zobrazenie, ktoré spomínanej multifunkcii priradí novú multifunkciu, ktorej riadky sú uzavreté pomocou zloženia Galoisových konexií druhého kontextu. Postup ako sa robí ľavý uzáver môžete pozorovať na Obr. 7, ktorý aplikuje znalosti obsiahnuté v definícii 3.4.



Obr. 7 Ľavý uzáver L -bondu β

3.2.2 Generátory horných susedov

V nasledujúcej definícii je uvedený postup vytvorenia novej multifunkcie z puta β a to takým spôsobom, že nejaký riadok/stĺpec nahradíme novým horným susedom intentu/extentu. Spomínané intenty a extenty získavame pomocou Bělohávkovho algoritmu (viď článok [2]), ktorý na výstupe vracia okrem množín intentov a extentov aj informáciu o ich horných/dolných susedoch, ktorá sa využíva v nasledujúcej metóde.

Definícia 3.5. Nech $\beta \in L\text{-Bonds}(C_1, C_2)$, $x_1 \in X_1$, $y_2 \in Y_2$, A je extent kontextu C_1 a B je intent kontextu C_2 . Definujme nové multifunkcie:

- 1) $m_1(\beta, x_1, B): X_1 \rightarrow L^{Y_2}$, taká že
 - a) $m_1(\beta, x_1, B)(b_1) = \beta(b_1)$, pre všetky $b_1 \in X_1: b_1 \neq x_1$
 - b) $m_1(\beta, x_1, B)(x_1) = B$
- 2) $m_2(\beta, y_2, A): Y_2 \rightarrow L^{X_1}$, taká že
 - a) $m_2(\beta, y_2, A)(a_2) = \beta^T(a_2)$, pre všetky $a_2 \in Y_2: a_2 \neq y_2$
 - b) $m_2(\beta, y_2, A)(y_2) = A$

Predošlé znalosti o uzávere a vytváraní novej multifunkcie sme použili pre vytvorenie nasledujúcich funkcií. Funkcia **MOVES₁**(β, x_1) generuje množinu všetkých prijateľných horných susedov vstupného L -bondu β nahradením intentu z kontextu C_2 ako pravdivostnú hodnotu vstupného objektu x_1 horným susedom zo zväzu všetkých intentov kontextu C_2 a analogicky funkcia **MOVES₂**(β, y_2) generuje množinu všetkých prijateľných horných susedov vstupného L -bondu β substituovaním extentu z kontextu C_1 ako pravdivostnú hodnotu vstupného atribútu y_2 horným susedom zo zväzu všetkých extentov kontextu C_1 .

Function MOVES₁(β, x_1)

```

1 UpperNeighborsOfBeta :=  $\emptyset$ 
2 UpperNeighborsOfIntent := NEIGHBORS( $\beta(x_1)$ )
3 foreach  $B \in \textit{UpperNeighborsOfIntent}$  do
4    $\gamma := \text{leftclosure}(m_1(\beta, x_1, B))$ 
5   if  $\gamma \in \text{Bonds}(\langle X_1, Y_1, I_1 \rangle, \langle X_2, Y_2, I_2 \rangle)$  then
6     add  $\gamma$  to UpperNeighborsOfBeta
7 end
8 return UpperNeighborsOfBeta

```

Function MOVES₂(β, y_2)

```

1 UpperNeighborsOfBeta :=  $\emptyset$ 
2 UpperNeighborsOfExtent := NEIGHBORS( $\beta^T(y_2)$ )
3 foreach  $A \in \textit{UpperNeighborsOfExtent}$  do
4    $\delta := \text{rightclosure}(m_2(\beta^T, y_2, A))$ 
5   if  $\delta \in \text{Bonds}(\langle X_1, Y_1, I_1 \rangle, \langle X_2, Y_2, I_2 \rangle)$  then
6     add  $\delta$  to UpperNeighborsOfBeta
7 end
8 return UpperNeighborsOfBeta

```

3.2.3 Generovanie horných susedov

Myšlienka generovania horných susedov je aplikovaná vo funkcii **NEIGHBORS**, ktorá akceptuje ako vstup L -bond β a vráti množinu všetkých horných susedov L -bondu na vstupe. Algoritmus používa premenné U , Min_1 a Min_2 . U je množina všetkých horných susedov, ktorá je na začiatku prázdna, Min_1 je množina objektov priestoru X a Min_2 je množina atribútov priestoru Y , ktoré vnímame ako množiny možných generátorov horných susedov vstupného L -bondu. Tieto množiny sú na začiatku naplnené prvkami, ktoré vo vstupnom L -bonde neukazujú na plné množiny. Robíme to kvôli tomu, že plné množiny v úplnom zväze nemajú horného suseda. Počas behu algoritmu sú z týchto množín odstránené tie prvky, ktoré nie sú generátormi horných susedov vstupného L -bondu. Na konci algoritmu sa v nich nachádzajú iba hľadané generátory. Ako je možné vidieť z riadkov 5, 10, 12 a 16, 21, 23 vo funkcii **NEIGHBORS**, prvok $b_1(a_2)$ ostane v množine Min_1 (Min_2) práve vtedy, ak L -bond $\gamma(\delta)$ je pridaný do množiny U . Množina *Objects* uchováva objekty b_1 priestoru X , pre ktoré platí, že L -množina $\beta(b_1)$ je podmnožinou L -množiny $\gamma(b_1)$, a podobne množina *Attributes* uchováva atribúty a_2 priestoru Y , pre ktoré platí, že L -množina $\beta^T(a_2)$ je podmnožinou L -množiny $\delta(a_2)$ (viď riadok 7 a 18 funkcie **NEIGHBORS**). Testy na riadkoch 9 a 20 funkcie **NEIGHBORS** sú úspešné práve vtedy, ak L -bond $\gamma(\delta)$ je horným susedom vstupného L -bondu, pričom prvok $b_1(a_2)$ je generátorom L -bondu $\gamma(\delta)$. Preto prvok $b_1(a_2)$ môžeme nechať v množine Min_1 (Min_2) a L -bond $\gamma(\delta)$ je pridaný do výslednej množiny U .

Function NEIGHBORS(β)

```
1  $U := \emptyset$ ;  
2  $Min_1 := \{x_1 \in X_1 | \beta(x_1) \neq Y_2\}$ ;  
3  $Min_2 := \{y_2 \in Y_2 | \beta^T(y_2) \neq X_1\}$ ;  
4 foreach  $x_1 \in X_1$  such that  $\beta(x_1) \neq Y_2$  do  
5   foreach  $\gamma \in \mathbf{MOVES}_1(\beta, x_1)$  do  
6      $Objects := \emptyset$   
7     foreach  $b_1 \in X_1; b_1 \neq x_1; \beta(b_1) \subset \gamma(b_1)$   
8        $Objects := Objects \cup \{b_1\}$   
9     if  $Min_1 \cap Objects = \emptyset$  then  
10       $U := U \cup \{\gamma\}$   
11    else  
12       $Min_1 := Min_1 - \{x_1\}$   
13    end  
14 end  
15 foreach  $y_2 \in Y_2$  such that  $\beta^T(y_2) \neq X_1$  do  
16   foreach  $\delta \in \mathbf{MOVES}_2(\beta, y_2)$  do  
17      $Attributes := \emptyset$   
18     foreach  $a_2 \in Y_2; a_2 \neq y_2; \beta^T(a_2) \subset \delta(a_2)$   
19        $Attributes := Attributes \cup \{a_2\}$   
20     if  $Min_2 \cap Attributes = \emptyset$  then  
21        $U := U \cup \{\delta^T\}$   
22     else  
23        $Min_2 := Min_2 - \{y_2\}$   
24     end  
25 end  
26 return  $U$ 
```

Funkcia **NEIGHBORS** vráti množinu všetkých horných susedov po konečnom počte krokov a jej konečnosť je zaručená vďaka redukcii množín Min_1 a Min_2 .

3.3 Algoritmus pre hľadanie všetkých pút

Hlavnou myšlienkou prezentovaného algoritmu je vybudovať každý L -bond β ako L -množinu objektov z priestoru X_1 s pravdivostnými stupňami z úplného zväzu všetkých intentov kontextu C_2 a analogicky pre transpozíciu β^T ako L -množinu atribútov z priestoru Y_2 s pravdivostnými stupňami z úplného zväzu všetkých extentov kontextu C_1 . Nami navrhnutý algoritmus je inšpirovaný algoritmom od tímu okolo Bělohlávka [2], ktorý vyhľadáva množinu všetkých L -konceptov, tak že každý prvok má svoju množinu horných a dolných susedov. Analogicky náš algoritmus vracia množinu všetkých L -bondov s množinami horných a dolných susedov.

Algoritmus začína uzáverom najmenšieho L -bondu a pridá ho do množiny nájdených pút. Najmenší L -bond vznikne ľavým uzáverom relácie, ktorá po riadkoch obsahuje najmenšie intenty druhého kontextu. Pre každý novo pridaný L -bond zistíme jeho horných susedov použitím funkcie **NEIGHBORS** a potom aktualizujeme informácie o dolných susedoch. Pre každého horného suseda, ktorý nebol nájdený v predošlých krokoch algoritmu rekurzívne opakujeme proces, kým nenarazíme na najväčší L -bond, t.j. plná relácia. Algoritmus rozdelíme do 2 funkcií: **LATTICEOFBONDS** a **GENERATEFROM**. Funkcia **LATTICEOFBONDS** akceptuje ako vstupy dva kontexty, medzi ktorými chceme nájsť všetky putá a iniciuje rekurzívne generovanie horných susedov. Funkcia **GENERATEFROM** generuje všetkých horných susedov vstupného L -bondu β a vytvára medzi nimi prepojenia horný/dolný sused. Je volaná rekurzívne pre každého horného suseda vstupného L -bondu β .

Function LATTICEOFBONDS(C_1, C_2)

```
1 Bonds :=  $\emptyset$ ;
2 foreach  $x_1 \in X_1$  do
3    $\beta(x_1) := \text{cl}_2(\emptyset)$ ;
4  $\beta := \text{leftclosure}(\beta)$ ;
5 add  $\beta$  to Bonds;
6 GENERATEFROM( $\beta$ );
7 return  $\langle \text{Bonds}, \{\beta^* | \beta \in \text{Bonds}\}, \{\beta_* | \beta \in \text{Bonds}\} \rangle$ 
```

Function GENERATEFROM(β)

```
1  $UpperNeighbors(\beta) := \text{NEIGHBORS}(\beta)$ 
2  $N := UpperNeighbors(\beta) - \text{Bonds}$ 
3 foreach  $\gamma \in UpperNeighbors(\beta)$  do
4   add  $\gamma$  to  $LowerNeighbors(\gamma)$ 
5   if  $\gamma \in N$  then
6     add  $\gamma$  to Bonds
7 end
8 foreach  $\gamma \in N$  do
9   GENERATEFROM( $\gamma$ )
10 end
```

3.4 Príklad výpočtu

V tejto časti si ukážeme príklad výpočtu na názornom príklade. Pre vstup nášho algoritmu sme zvolili dva kontexty uvedené v Tab. 8 a Tab. 9.

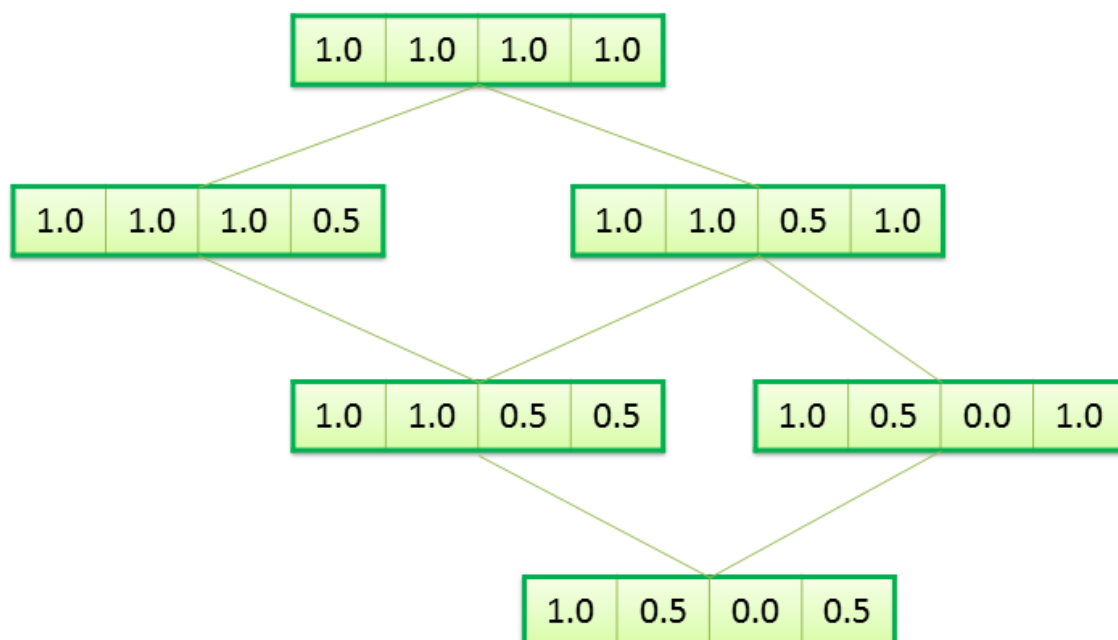
Tab. 8 Prvý *L*-kontext ilustračného príkladu

0.0	1.0	0.5	0.5	0.5
0.0	0.5	0.5	1.0	1.0

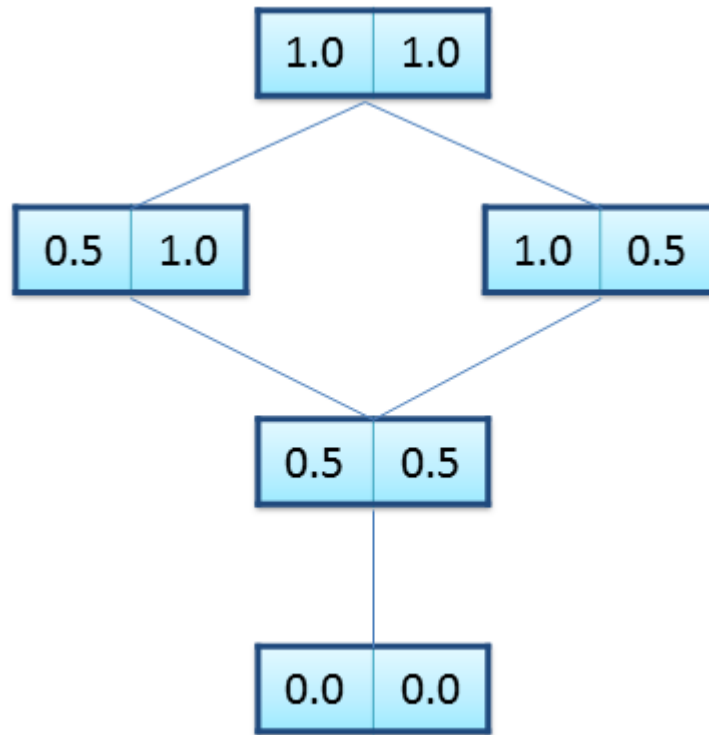
Tab. 9 Druhý *L*-kontext ilustračného príkladu

1.0	1.0	0.5	1.0
1.0	1.0	1.0	0.5
1.0	0.5	0.0	1.0

Na začiatku potrebujeme zistiť konceptové zväzy vstupných kontextov, aby sme mali informácie o horných susedoch každého konceptu, ktoré sa používajú pri generovaní horných susedov daného puta. Na tieto účely použijeme algoritmus z predchádzajúcej kapitoly. Pre jednoduchosť a potreby nášho algoritmu nám postačuje úplný zväz intentov druhého kontextu a úplný zväz extentov prvého kontextu. Znázorňujeme ich na obrázkoch Obr. 8 a Obr. 9.



Obr. 8 Úplný zväz intentov druhého *L*-kontextu



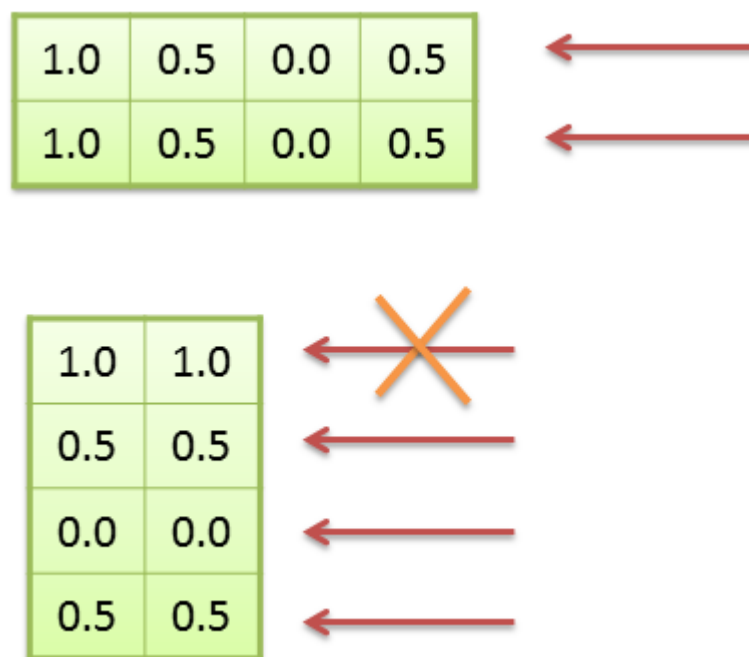
Obr. 9 Úplný zväz extentov prvého L -kontextu

Algoritmus začína uzáverom najmenšieho L -bondu a pridá ho do množiny nájdených pút. Najmenší L -bond vznikne ľavým uzáverom relácie (viď definícia 3.4.), ktorá po riadkoch obsahuje najmenšie intenty druhého kontextu. Pokračujeme jednou názornou iteráciou funkcie **NEIGHBORS**, ktorá nájde všetkých horných susedov najmenšieho L -bondu. V nasledujúcom budeme striedať úryvky pseudokódu a podrobnejší popis konkrétnych myšlienok v algoritme.

$$Min_1 := \{x_1 \in X_1 | \beta(x_1) \neq Y_2\};$$

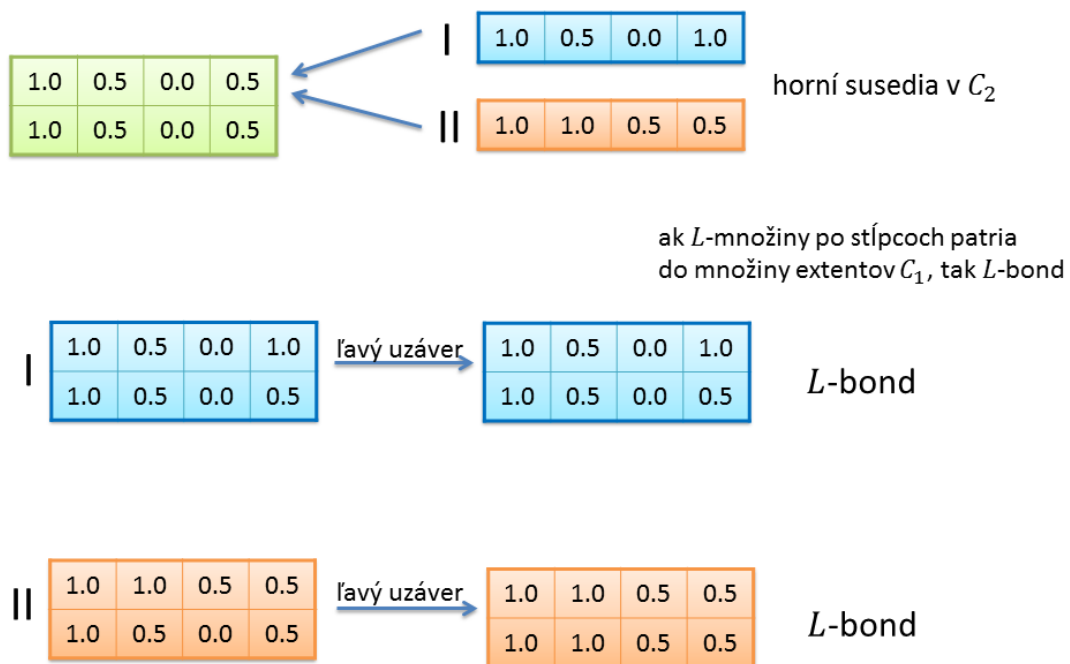
$$Min_2 := \{y_2 \in Y_2 | \beta^T(y_2) \neq X_1\};$$

Začneme výberom iba tých objektov/atribútov, pre ktoré L -bond β nepriradí plnú L -množinu. Faktom je, že takáto L -množina je najväčšou v úplnom zväze, a tým pádom nemá ďalšieho horného suseda a je zbytočné sa ňou zaoberať. Z Obr. 10 vyplýva, že ktoré objekty/atribúty budú vybrané pre ďalšie spracovanie: $Min_1 := \{0,1\}$ a $Min_2 := \{1,2,3\}$.



Obr. 10 Objekty a atribúty L -bondu skúmané vo funkcii NEIGHBORS

Funkciou **MOVES₁** vygenerujeme množinu všetkých prijateľných horných susedov L -bondu β nahradením intentu z kontextu C_2 ako pravdivostnú hodnotu vstupného objektu x_1 horným susedom zo zväzu všetkých intentov kontextu C_2 . Podobne, funkciou **MOVES₂** vygenerujeme množinu všetkých prijateľných horných susedov L -bondu β substituovaním extentu z kontextu C_1 ako pravdivostnú hodnotu vstupného atribútu y_2 horným susedom zo zväzu všetkých extentov kontextu C_1 . Na Obr. 11 môžete pozorovať ako prebieha funkcia **MOVES₁** pre náš najmenší L -bond a prvý objekt priestoru X_1 .



Obr. 11 Pribeh funkcie MOVES_1 pre najmenší L -bond a prvý objekt priestoru X_1

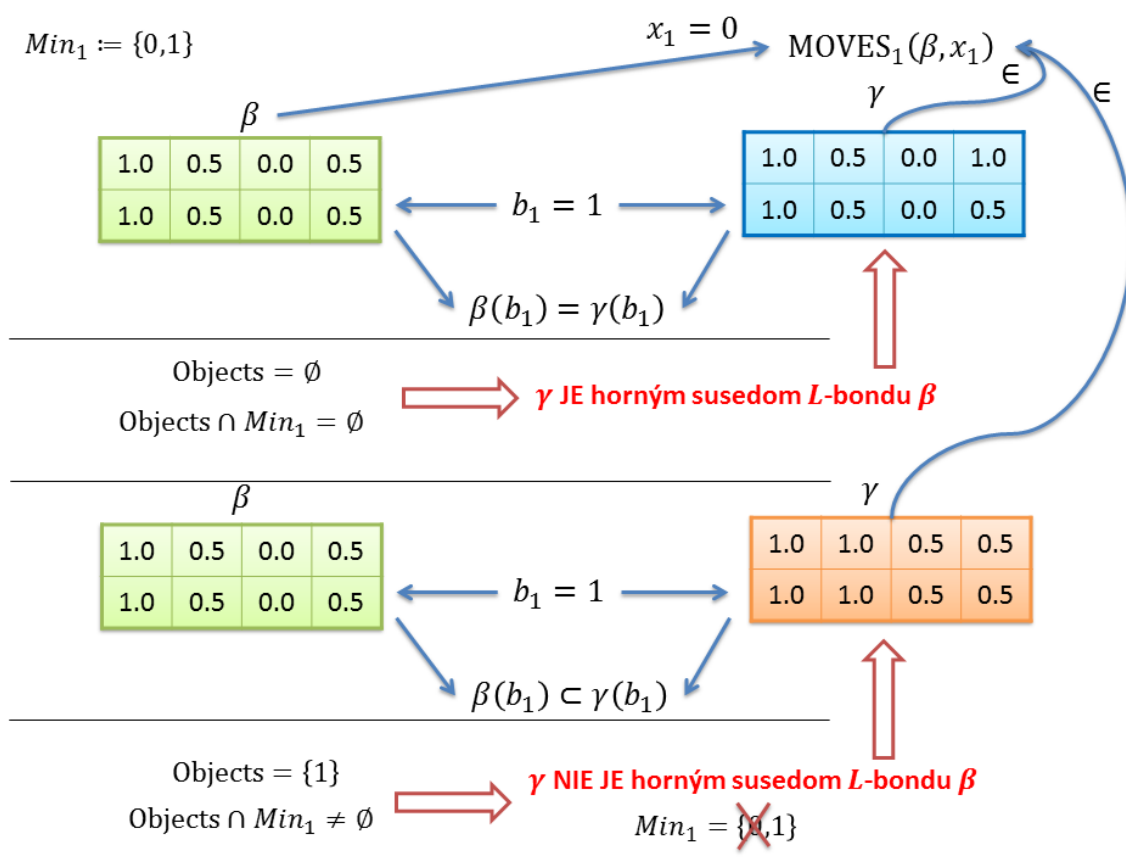
Akonáhle sme zistili množinu všetkých možných prijateľných horných susedov najmenšieho L -bondu β , môžeme začať vyberať tie, ktoré sú reálne hornými susedmi. Na riadku 7 časti pseudokódu funkcie **NEIGHBORS** vidíme kritériá výberu.

```

5  foreach  $\gamma \in \text{MOVES}_1(\beta, x_1)$  do
6     $Objects := \emptyset$ 
7    foreach  $b_1 \in X_1; b_1 \neq x_1; \beta(b_1) \subset \gamma(b_1)$ 
8       $Objects := Objects \cup \{b_1\}$ 
9    if  $Min_1 \cap Objects = \emptyset$  then
10      $U := U \cup \{\gamma\}$ 
11  else
12      $Min_1 := Min_1 - \{x_1\}$ 
13 end

```

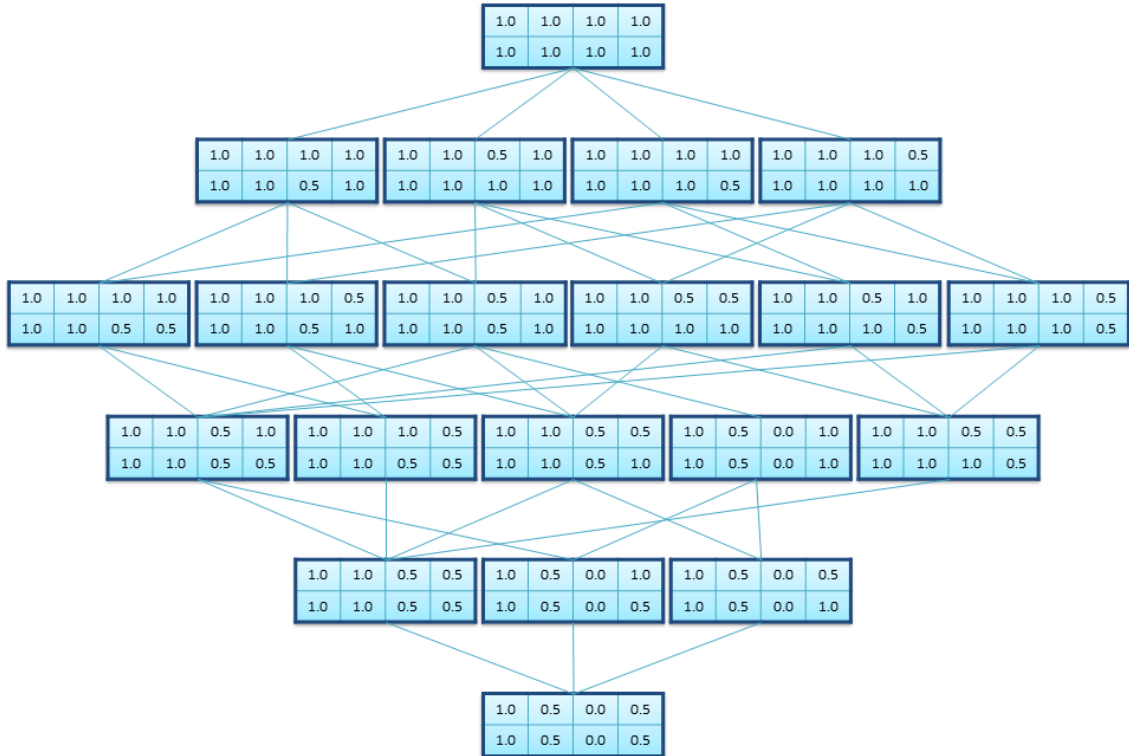
Vyberáme objekty, ktoré sú rôzne od práve spracovávaného objektu x_1 (nám sme získali množinu prijateľných horných susedov L -bondu β funkciou $\mathbf{MOVES}_1(\beta, x_1)$) a tie, pre ktoré L -množina $\beta(b_1)$ je podmnožinou L -množiny $\gamma(b_1)$. Ak sú podmienky splnené, tak objekt b_1 pridáme do množiny indexov $Objects$. Ak v aktuálnej iterácii je prienik množín Min_1 a $Objects$ prázdny, tak L -bond γ je horným susedom L -bondu β . Inak práve spracovávaný objekt x_1 odstránime z množiny Min_1 . Priebeh časti funkcie **NEIGHBORS**, kde rozhodujeme, či daný možný horný sused je reálnym horným susedom môžete pozorovať na Obr. 12. Keďže L -bond β je definovaný aj transpozíciou, rovnaký postup použijeme aj z pohľadu atribútov druhého kontextu.



Obr. 12 Priebeh časti funkcie **NEIGHBORS**, kde vyberáme reálnych horných susedov L -bondu β

Výstupom funkcie **NEIGHBORS** je množina horných susedov vstupného L -bondu β . V našom príklade pre najmenší L -bond β sme našli troch horných susedov (viď Obr. 13). Aby sme vybudovali úplný zväz L -bondov, potrebujeme nejakú rekurzívnu funkciu, ktorá sa zastaví keď vygeneruje najväčší L -bond. Túto záležitosť

rieši funkcia **GENERATEFROM**, ktorá generuje všetkých horných susedov vstupnému L -bondu β a vytvára medzi nimi prepojenia horný/dolný sused. Je volaná rekurzívne pre každého horného suseda vstupného L -bondu β . Nakoniec sa funkcia **GENERATEFROM** zavolá pre najmenší L -bond β vo funkcii **LATTICEOFBONDS**, ktorá sa stará o začiatkové inicializácie premenných a vytvorenie najmenšieho L -bondu β . Výstupom funkcie **LATTICEOFBONDS** pre L -kontexty uvedené v Tab. 8 a Tab. 9 je úplný zväz L -bondov na Obr. 13.

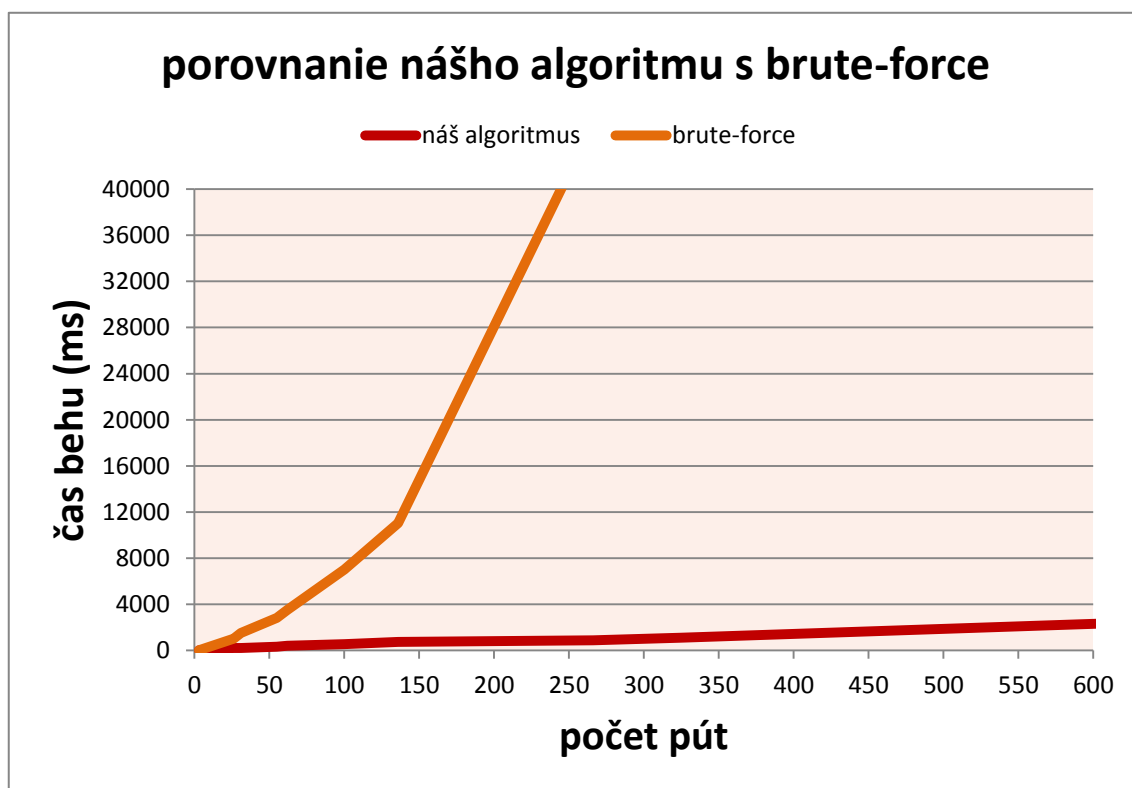


Obr. 13 Úplný zväz všetkých L -bondov

3.5 Porovnanie nášho algoritmu s brute-force prístupom

Vykonalí sme sériu experimentov s náhodne generovanými fuzzy kontextami, pričom pre účely generovania sme použili trojhodnotovú Lukasiewiczovu logiku. „**Fill ratio**“ týchto kontextov, t.j. percentuálny podiel nenulových položiek v kontexte, bolo na úrovni 25%, 50%, 75% a 100%. Dôvod pre použitie „fill ratia“ pri generovaní bol fakt, že čím menej nenulových položiek v kontextoch, tým menej pút. Vygenerované kontexty sme následne použili ako vstup pre brute-force a náš efektívny algoritmus. Všetky algoritmy, generovanie dát a testy boli implementované v programovacom

jazyku Java. Na Obr. 14 je znázornená závislosť času behu algoritmov na veľkosti hľadaného zväzu pút.



Obr. 14 Závislosť času behu algoritmov na veľkosti hľadaného zväzu pút

4 Pseudoriešenie 9. otvoreného problému

V roku 2006 bolo na konferencii ICFCA v Drážďanoch stanovených a zoskupených v zbierke [5] niekoľko otvorených problémov pre formálnu konceptovú analýzu, z ktorých deviaty sa nazýva „porozumenie“ pút. Znenie problému by sa dalo stručne formulovať ako jednoduchá konštrukcia kontextu (prirodzených rozmerov), ktorého konceptový zväz je izomorfný so zväzom všetkých pút medzi danými kontextami, t.j. majú rovnaký počet prvkov a hierarchickú štruktúru.

Jedným z cieľov diplomovej práce bolo prispieť k jeho riešeniu. Finálnym výstupom nášho snaženia je „pseudoriešenie“, ktoré hľadaný kontext konštruje zložitejšou cestou. Predpona pseudo naznačuje neoptimálnosť nášho riešenia v takom zmysle, že na konštrukciu hľadaného kontextu je potrebné najskôr vyhľadať všetky putá medzi vstupnými kontextami, ktorých je vo všeobecnosti veľa. V nasledujúcej časti si predstavíme spôsob ako z úplného zväzu všetkých pút medzi dvoma kontextami získať hľadaný kontext.

4.1 Transformácia zväzu pút na prirodzene malý kontext

V tejto časti si ukážeme ako z množiny všetkých pút vybrať iba dôležité putá pomocou jednoduchých faktov z teórie zväzov obsiahnutých v druhej časti základnej vety pre konceptové zväzy. Pre pripomenutie ju uvádzame:

Veta 4.1. Úplný zväz V je izomorfný s konceptovým zväzom $CL(X, Y, I)$ práve vtedy, keď existujú zobrazenia $\varphi: X \rightarrow V$ a $\phi: Y \rightarrow V$ také, že $\varphi(X)$ je suprénum hustá množina vo zväze V , $\phi(Y)$ je infimum hustá množina vo zväze V a $I(x, y)$ je ekvivalentné $\varphi(x) \leq \phi(y)$, pre všetky objekty $x \in X$ a atribúty $y \in Y$. To znamená, že V je izomorfný s konceptovým zväzom $\beta(V, V, \leq)$.

Z poslednej poznámky predchádzajúcej vety vyplýva, že triviálnym riešením by mohol byť kontext, ktorého objektová i atribútová množina je rovná množine všetkých pút medzi vstupnými kontextami a relácia daného kontextu by bola relácia usporiadania daných pút. Konceptový zväz takéhoto kontextu je izomorfný so zväzom všetkých pút, pričom ale nespĺňa podmienku prirodzenej jednoduchosti (triviálny kontext je

rozmerovo príliš veľký). V nasledujúcich definíciach uvedených v knihe [1] si objasníme spomenuté pojmy.

Definícia 4.1. Nech V je úplný zväz. Pre prvok v zväzu V definujeme:

$$v_* = \bigvee \{x \in V \mid x < v\}$$

$$v^* = \bigwedge \{x \in V \mid x > v\}$$

Ak platí $v_* \neq v$, tak prvok v nazývame **suprémum ireducibilný**, t.j. prvok v nemôže byť vyjadrený ako suprémum od neho menších prvkov. V tomto prípade, prvok v_* je jediným dolným susedom prvku v . Analogicky, ak platí $v^* \neq v$, tak prvok v nazývame **infimum ireducibilný**, t.j. prvok v nemôže byť vyjadrený ako infimum od neho väčších prvkov. V tomto prípade, prvok v^* je jediným horným susedom prvku v .

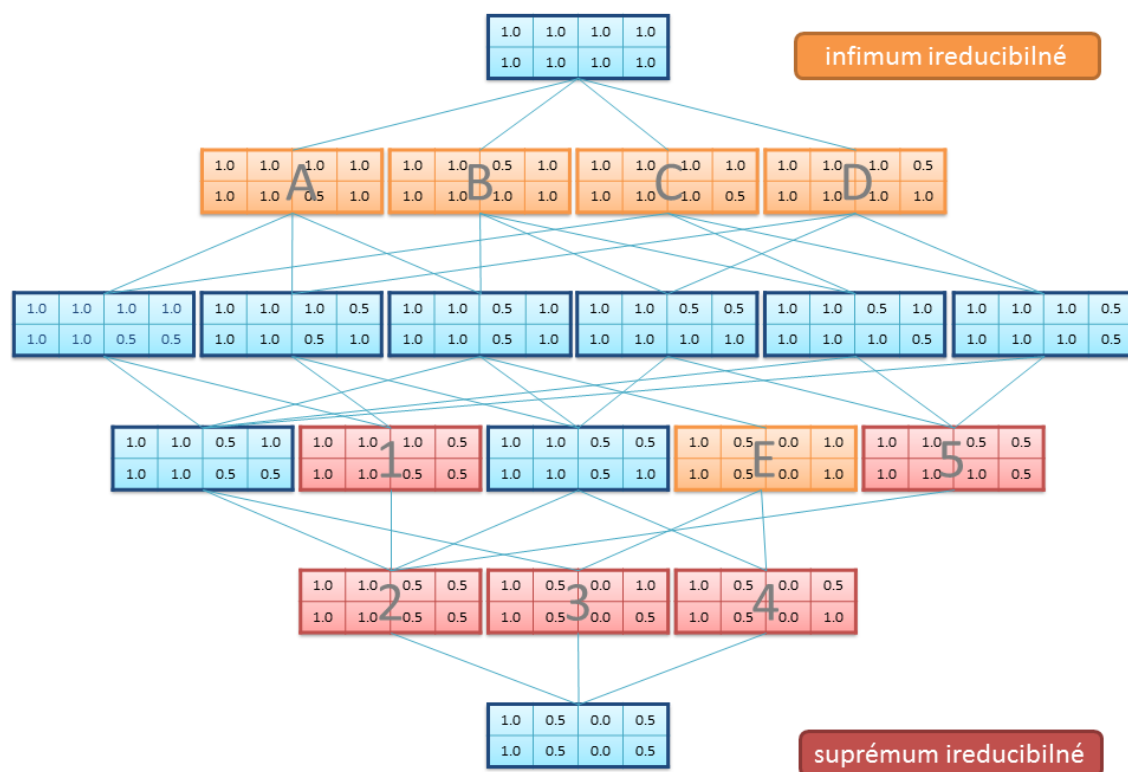
Definícia 4.2. Nech V je úplný zväz. Množinu $X \subseteq V$ nazývame **suprémum hustá** v zväze V , ak každý prvok zväzu V môžeme vyjadriť ako suprémum nejakej podmnožiny množiny X . Analogicky množinu $X \subseteq V$ nazývame **infimum hustá** v zväze V , ak každý prvok zväzu V môžeme vyjadriť ako infimum nejakej podmnožiny množiny X .

Nasledujúca veta opisuje jednoduchý spôsob ako z akéhokoľvek úplného zväzu (teda aj zväzu všetkých pút medzi dvoma kontextami) získať suprémum a infimum husté množiny.

Veta 4.2. Prvok v zväzu V je suprémum ireducibilný práve vtedy, keď má práve jedného dolného suseda. Prvok v zväzu V je infimum ireducibilný práve vtedy, keď má práve jedného horného suseda. Každá suprémum hustá podmnožina zväzu V obsahuje všetky suprémum ireducibilné prvky zväzu V a každá infimum hustá podmnožina zväzu V obsahuje všetky infimum ireducibilné prvky zväzu V . Naopak, množina všetkých suprémum ireducibilných prvkov tvorí suprémum hustú množinu. Analogicky, množina všetkých infimum ireducibilných prvkov tvorí infimum hustú množinu.

Nový kontext vytvoríme tak, že za množinu objektov prehlásime množinu suprémum ireducibilných prvkov, za množinu atribútov prehlásime množinu infimum ireducibilných prvkov v zväze všetkých pút a relácia medzi nimi bude relácia usporiadania L -multifunkcií. Na Obr. 15 sú zvýraznené všetky infimum a suprémum

ireducibilné prvky. Ich vzájomná relácia bola odzrkadlená v prirodzene malom kontexte v Tab. 10, ktorého konceptový zväz na Obr. 16 je izomorfný so zväzom všetkých L -bondov. Inak povedané, obidva zväzy majú rovnaký počet prvkov a hierarchiu.

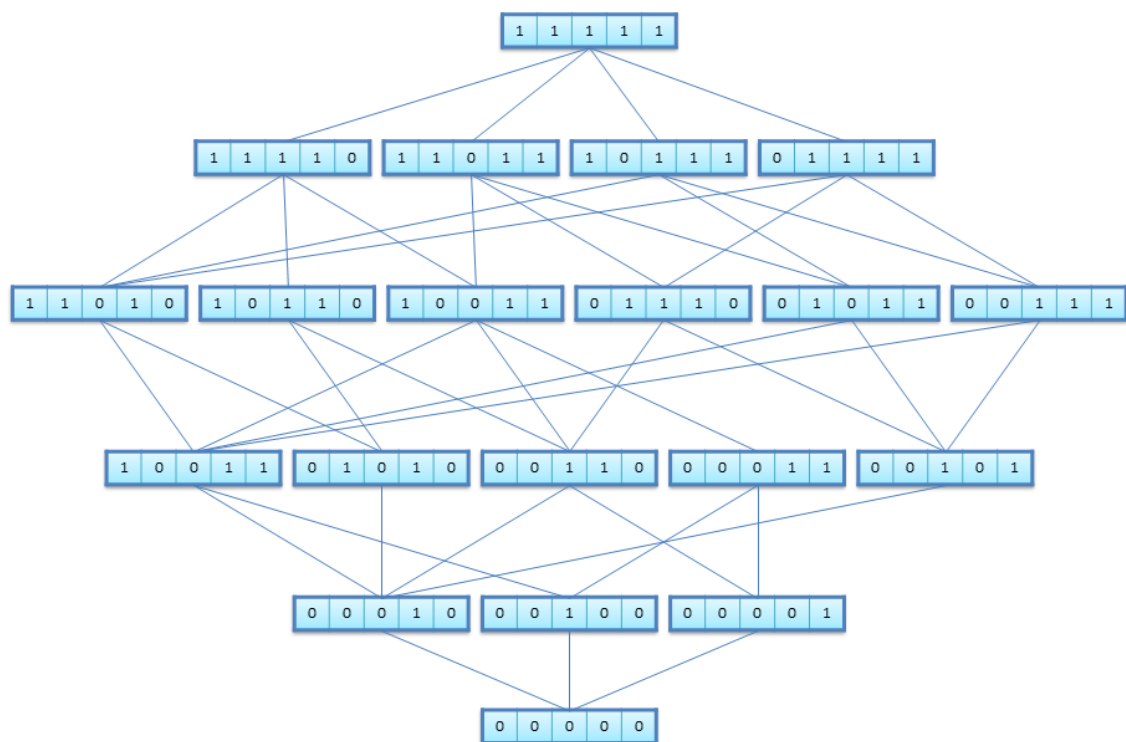


Obr. 15 Zväz všetkých L -bondov a zvýraznené všetky infimum a suprérum ireducibilné prvky

Tab. 10 Prirodzene malý kontext

infimum ireducibilné prvky

suprérum ireducibilné prvky	$\leq$	A	B	C	D	E
	1	1.0	0.0	1.0	0.0	1.0
	2	0.0	0.0	1.0	1.0	1.0
	3	1.0	1.0	0.0	1.0	1.0
	4	1.0	0.0	1.0	1.0	1.0
	5	1.0	1.0	1.0	1.0	0.0



Obr. 16 Úplný zváz extentov prirodzene malého kontextu

5 Spojitosť medzi putami a extentami priameho súčinu dvoch formálnych kontextov

V tejto kapitole si ukážeme jednoduchý spôsob získavania pút dvoch kontextov, ktorého nevýhodou je, že výstupom je neúplná množina všetkých pút. Existuje spojitosť medzi extentami priameho súčinu kontextov a putami, t.j. každý extent priameho súčinu kontextov je automaticky puto (viď článok [4]). Poďme si formálne zaviesť priamy súčin kontextov definíciou z článku [4]:

Definícia 5.1. Priamym súčinom dvoch L -kontextov $\langle X_1, Y_1, I_1 \rangle$ a $\langle X_2, Y_2, I_2 \rangle$ je nový L -kontext $\langle X_1 \times Y_2, Y_1 \times X_2, \Delta \rangle$, kde relácia Δ je definovaná nasledovne:

$$\Delta((x_1, y_2), (y_1, x_2)) = \neg I_1(x_1, y_1) \rightarrow I_2(x_2, y_2)$$

Jednoduchým faktom z logiky je, že disjunkcia dvoch pravdivostných hodnôt a, b môže byť vyrátaná ekvivalentne ako $a \text{ OR } b = \neg a \rightarrow b$, pričom ale komutatívnosť disjunkcie je vynútená pravidlom dvojitej negácie $\neg\neg a = a$, pre každú hodnotu a .

Veta 5.1. Nech L je reziduovaný zväz (splňajúci aj pravidlo dvojitej negácie). Nech $C_i = \langle X_i, Y_i, I_i \rangle$ sú L -kontexty pre $i \in \{1, 2\}$. Potom pre každý extent β priameho súčinu $C_1 \Delta C_2$ platí, že β je puto vstupných kontextov.

Na Obr. 17 vidíme priamy súčin kontextov z nášho príkladu a na Obr. 18 sú znázornené putá nájdené pomocou priameho súčinu. Je zrejmé, že množina pút nájdených pomocou priameho súčinu je podmnožinou množiny všetkých pút. Dôvodom je nevygenerovanie všetkých supremum/infimum ireducibilných pút.

$$C_1 = \langle X_1, Y_1, I_1 \rangle$$

0.0	1.0	0.5	0.5	0.5
0.0	0.5	0.5	1.0	1.0

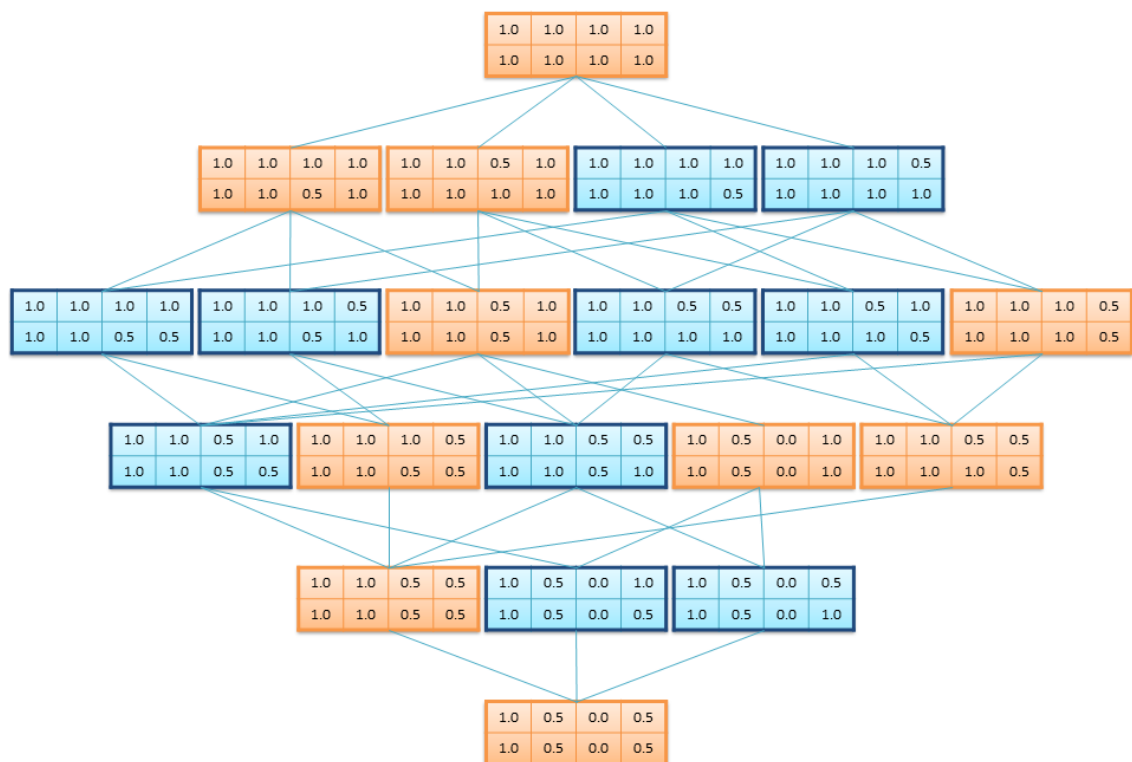
$$C_2 = \langle X_2, Y_2, I_2 \rangle$$

1.0	1.0	0.5	1.0
1.0	1.0	1.0	0.5
1.0	0.5	0.0	1.0

$$C = \langle X_1 \times Y_2, Y_1 \times X_2, \Delta \rangle = C_1 \Delta C_2$$

1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0	1.0	1.0
0.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.5	0.5	0.5
1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0	1.0	1.0
0.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.5	0.5	1.0	1.0
1.0	1.0	1.0	1.0	1.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Obr. 17 Priamy súčin kontextov C_1 a C_2

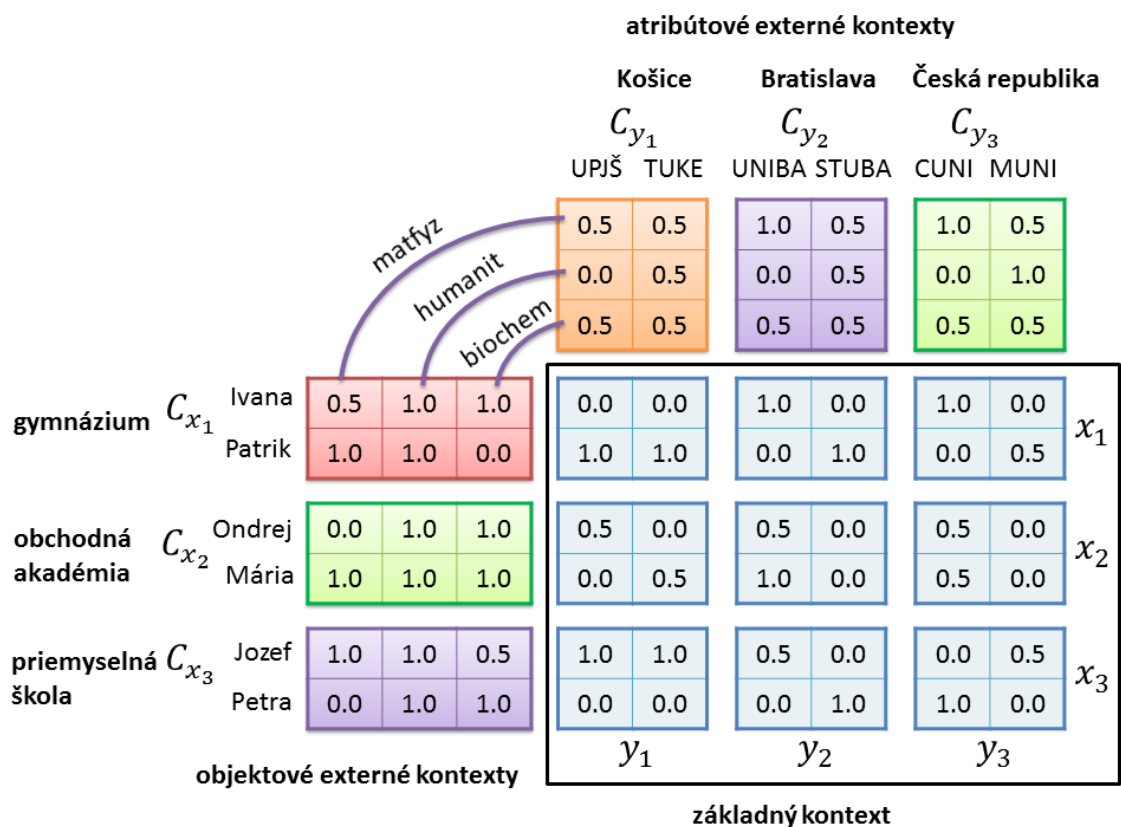


Obr. 18 Úplný zväz všetkých pút a zvýraznené putá nájdené pomocou priameho súčinu

6 Využitie pút

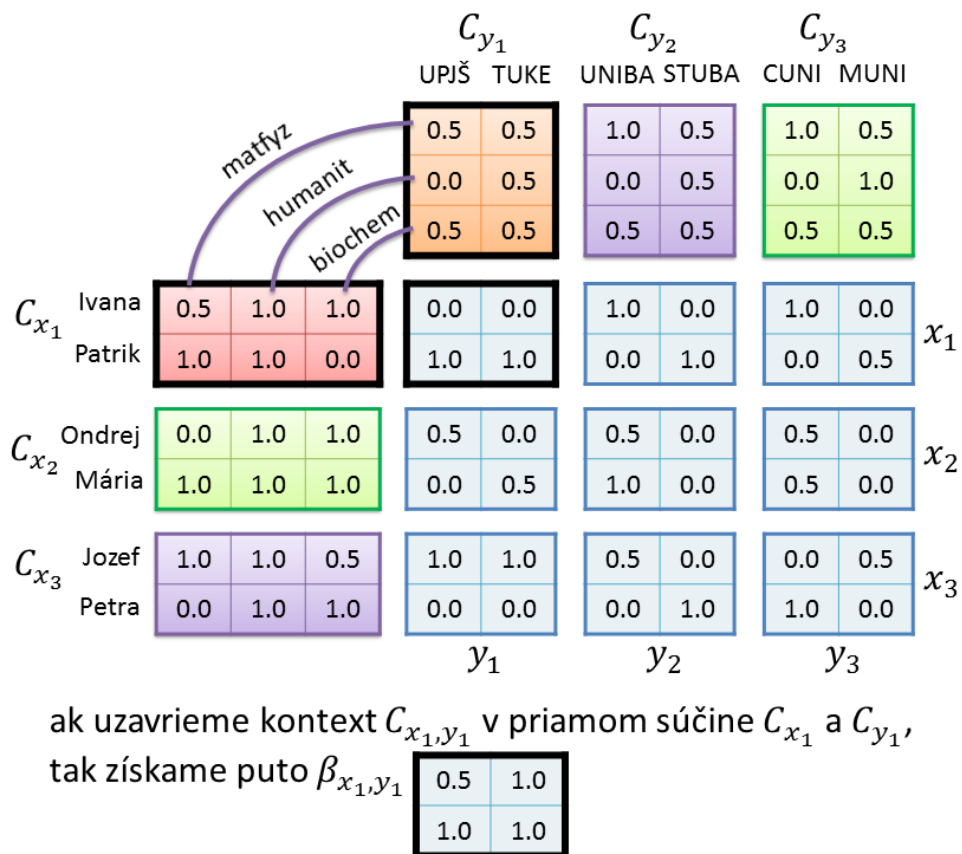
V poslednej časti diplomovej práce sa venujeme využitiu pút pri konštrukcii úplného zväzu vyššieho rádu. Naše poznatky o putách sme aplikovali na zaujímavý príklad z praxe.

Predstavte si, že ste študentom strednej školy a nastal moment, kedy musíte urobiť dôležité rozhodnutie, t.j. vybrať si univerzitu, na ktorej budete študovať. Majme k dispozícii informácie o študentoch stredných škôl a ich postojoch k univerzitám, ktoré majú na výber, konkrétne v našom prípade šesť študentov stredných škôl a šesť univerzít na výber. Z týchto informácií sme zostavili **základný kontext**, kde objektovú časť tvoria študenti a atribútovú časť tvoria univerzity. Ďalej majme k dispozícii ďalšie **externé dáta** od stredných škôl a univerzít, konkrétne hodnotenia študentov v okruhoch predmetov (matematika - fyzika, humanitné predmety a biológia - chémia) a požiadavky univerzít na tie isté okruhy predmetov, ktoré rozhodujú o tom, či daný uchádzač o štúdium bude na univerzitu prijatý alebo nie. Získané externé dáta od stredných škôl utriedime do skupín podľa typu strednej školy, v našom prípade gymnázium, obchodná akadémia a stredná priemyselná škola a dáta od univerzít podľa sídla univerzity, v našom prípade Košice, Bratislava, Česká republika. Týmto rozdelením množiny študentov a množiny škôl do menších a disjunktných celkov (študentov podľa stredných škôl a univerzity podľa sídla) získavame aj rozdelenie základného kontextu preferencií študentov do menších celkov – podkontextov znázornených na Obr. 19.



Obr. 19 Základný kontext preferencií študentov a externé kontexty študijných výsledkov a požiadaviek univerzít

Na základe teoretických výsledkov z článku [10] je možné z puta medzi dvoma kontextami vyrobiť Galoisovu konexiu medzi konceptovými zväzmi vstupných kontextov. Po predchádzajúcom rozdelení základného kontextu môžeme uvažovať nad novým kontextom, ktorého objektmi sú jednotlivé podmnožiny pôvodných objektov a podobne atribútmi sú podmnožiny pôvodných atribútov. Akýkoľvek podkontext prislúchajúci nejakej objektovo-atribútovej dvojici môžeme vnímať ako pseudoputo medzi prislúchajúcim objektovým/atribútovým kontextom. Predpona pseudo znamená to, že prislúchajúci podkontext putom nie je, ale pomocou uzáveru, vyplývajúceho z priameho súčinu objektového a atribútového kontextu, aplikovaného na pseudoputo, jednoducho získame puto medzi spomínanými kontextami (viď Obr. 20).



Obr. 20 Vytvorenie puta medzi objektovým a atribútovým kontextom

Z tohto pohľadu sme získali menší kontext, ktorého každý objekt a každý atribút je možné ohodnotiť hodnotou z prislúchajúceho konceptového zväzu objektového alebo atribútového kontextu. Pokiaľ každé objektovo-atribútové puto vnímame ako Galoisovu konexiu medzi prislúchajúcimi konceptovými zväzmi, ktoré vystupujú ako štruktúry pravdivostných hodnôt prislúchajúceho objektu alebo atribútu, dostávame formálny kontext tvaru opísanom v článku [11], ktorého teoretické výsledky dokazujú korektnosť daného prístupu a dávajú návod ako daný kontext spracovať do formy konceptového zväzu. Konkrétne sme vytvorili špeciálne kontextové operátory, ktoré n -tici konceptov objektových kontextov priradujú m -tici konceptov atribútových kontextov a naopak.

V našom príklade mal základný kontext 51 konceptov a použitím externých dát sa počet konceptov znížil na 3. Je taktiež nutné poznamenať, že prístupom využívajúcim putá medzi kontextami sme spravodlivo priradili podobným uzavretým množinám študentov podobné uzavreté množiny škôl.

Záver

V tejto diplomovej práci sme sa na začiatku oboznámili so znalosťami, ktoré sú základnými stavebnými blokmi formálnej konceptovej analýzy a predstavili sme si tiež Bělohávkovo fuzzy rozšírenie formálnej konceptovej analýzy. Zaoberali sme sa efektívnym vyhľadávaním konceptov v kontexte a podrobne sme čitateľovi predostreli myšlienku hľadania všetkých uzavretých množín, tzv. fixpointov algoritmom, ktorý využíva paradigmy horných susedov.

V ďalšej časti sme pracovali na hľadaní významov medzi dvoma kontextami, inak povedané pút, čo sú relácie medzi priestorom objektov prvého kontextu a priestorom atribútov druhého kontextu, také že ich riadky a stĺpce sú uzavreté množiny v príslušnom kontexte. Navrhli sme algoritmus pre ich hľadanie, ktorý bol inšpirovaný algoritmom hľadania všetkých uzavretých množín. Dôvodom znovu použitia myšlienky „horných susedov“ bol fakt, že množina všetkých konceptov tvorí úplný zväz vzhľadom na množinovú inklúziu a množina všetkých pút tvorí úplný zväz vzhľadom na relačné usporiadanie.

Deviaty otvorený problém vo formálnej konceptovej analýze hľadá odpoveď na otázku hľadania prirodzene malého kontextu pre uzavretý systém pút medzi kontextami alebo inak povedané hľadania jednoduchej konštrukcie kontextu, ktorého konceptový zväz je izomorfný so zväzom všetkých pút medzi danými kontextami.

Táto skutočnosť bola motiváciou nášho výskumu, kde sme sa snažili prispieť k riešeniu tohto problému. Finálnym výstupom nášho snaženia je „pseudoriešenie“, ktoré hľadaný kontext konštruje zložitejšou cestou. Predpona pseudo naznačuje neoptimálnosť nášho riešenia v takom zmysle, že na konštrukciu hľadaného kontextu je potrebné najskôr vyhľadať všetky putá medzi vstupnými kontextami, ktorých je vo všeobecnosti veľa.

V nasledujúcej časti sme si predstavili spôsob ako z úplného zväzu všetkých pút medzi dvoma kontextami získať hľadaný kontext.

Keďže existuje spojitosť medzi extentami priameho súčinu kontextov a putami, t.j. každý extent priameho súčinu kontextov je automaticky puto, tak v poslednej fáze diplomovej práce sme sa venovali využitiu pút pri konštrukcii úplného zväzu vyššieho rádu. Naše poznatky o putách sme aplikovali na zaujímavý príklad z praxe, kde hľadáme skryté informácie vo viacerých spolu súvisiacich kontextoch.

Zoznam použitej literatúry

- [1] B. Ganter, R. Wille: Formal Concept Analysis, Mathematical Foundation, Springer Verlag 1999, ISBN 3-540-62771-5
- [2] R. Bělohlávek, B. De Baets, J. Outrata, V. Vychodil: Computing the lattice of all fixpoints of a fuzzy closure operator, IEEE Transactions On Fuzzy Systems, vol. 18, no.3, June 2010
- [3] O. Krídlo, S. Krajči, M. Ojeda-Aciego: The category of L-Chu correspondences and the structure of L-bonds, special issue of extended proceedings of CLA2010, Fundamenta Informaticae 115(4): 297-325 (2012)
- [4] O. Krídlo, S. Krajči, M. Ojeda-Aciego: L-bonds vs extents of direct products of two L-fuzzy contexts, In Proc. of Concept Lattices and Applications, CLA 2010, pages 70-79, 2010
- [5] Problems presented at ICFCA 2006 in Dresden
- [6] M. Krötzsch, P. Hitzler, G-Q. Zhang: Morphisms in Context, Lecture Notes in Computer Science 3596:223–237, 2005
- [7] R. Bělohlávek: Lattices of fixed points of fuzzy Galois connections, Mathematical Logic Quartely, 47(1):111–116, 2001
- [8] R. Bělohlávek: Concept lattices and order in fuzzy logic, Annals of Pure and Applied Logic, 128:277-298, 2004
- [9] P. Mihalčín: Faktorová analýza formálneho fuzzy kontextu, bakalárska práca (2011)
- [10] Ondrej Krídlo, Manuel Ojeda-Aciego: Linking L-Chu Correspondences and Completely Lattice L-Ordered Sets. CLA 2012: 245-257
- [11] J. Pócs: Note on generating fuzzy concept lattices via Galois connections, Information Sciences 185 (1), pp. 128-136, 2012

Prílohy

Príloha A: CD médium – diplomová práca v elektronickej podobe

Príloha B: CD médium – Java projekt, v ktorom sú naimplementované všetky algoritmy spomínané v práci