

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Program na podporu výuky uspořádaných množin



2018

Vedoucí práce: doc. RNDr. Mi-
roslav Kolařík, Ph.D.

Oldřich Beneš

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Oldřich Beneš
Název práce: Program na podporu výuky uspořádaných množin
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2018
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.
Počet stran: 33
Přílohy: 1 DVD
Jazyk práce: český

Bibliographic info

Author: Oldřich Beneš
Title: Program for support of teaching of ordered sets
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2018
Study field: Applied Computer Science, full-time form
Supervisor: doc. RNDr. Miroslav Kolařík, Ph.D.
Page count: 33
Supplements: 1 DVD
Thesis language: Czech

Anotace

V této bakalářské práci byla vytvořena aplikace na podporu výuky uspořádaných množin. Je určena pro studenty, kteří mají zájem rozšířit své znalosti v oblasti binárních relací na množině a uspořádaných množin. Součástí práce je výukový text.

Synopsis

In this bachelor's thesis was created an application for support of teaching partially ordered sets. This thesis is for students who are interested in expanding their knowledge about binary relation on a set and ordered sets. There is supporting educational text as the part of this thesis.

Klíčová slova: Binární relace, uspořádané množiny, supremum, infimum, svaz, Hasseův diagram.

Keywords: Binary relation, ordered sets, supremum, infimum, lattice, Hasse diagram.

Děkuji doc. RNDr. Miroslavu Kolaříkovi, Ph.D. za vedení této bakalářské práce, i za podnětné rady a připomínky při konzultacích. Rovněž bych rád poděkoval své rodině za trpělivost a podporu při studiu.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Teoretická část	7
1.1	Pojem binární relace	7
1.2	Vlastnosti binárních relací	8
1.3	Grafické znázornění relací	9
1.4	Relace uspořádání	10
1.5	Řetězec a antiřetězec	11
1.6	Relace pokrytí a Hasseovy diagramy	12
1.7	Speciální prvky uspořádaných množin	13
1.8	Dolní a horní kužel, supremum a infimum	15
1.9	Svazy a polosvazy	16
2	Implementace	18
2.1	Použité technologie	18
2.2	WPF	18
2.3	Použité knihovny	18
2.4	Reprezentace dat	19
2.5	MVVM	19
2.6	Struktura aplikace	20
2.7	Technika kreslení Hasseových diagramů a jejich využití ve výpočtech	20
2.8	Výpočty založené na Hasseových diagramech	21
2.9	Popis tříd	22
3	Uživatelská příručka	25
3.1	Požadavky	25
3.2	Instalace aplikace	25
3.3	Aplikace	25
3.3.1	Praktická část	25
3.3.1.1	Stránka Vlastnosti	26
3.3.1.2	Stránka Hasseovy diagramy	26
3.3.1.3	Stránka Speciální prvky	27
3.3.1.4	Stránka Kužely	27
3.3.1.5	Stránka Svazy a polosvazy	27
3.3.1.6	Okno Úprava množiny	28
3.3.2	Teoretická část	29
3.4	Odstalování aplikace	29
	Závěr	30
	Conclusions	31
4	Obsah přiloženého DVD	32
	Bibliografie	33

Seznam obrázků

1	Orientovaný graf pro relaci R_6	10
2	Postup vytvoření Hasseova diagramu pro (X, R_2)	12
3	Hasseův diagram pro (X, R_7)	13
4	Hasseův diagram pro $(P(X), \subseteq)$, kde $X = \{a, b, c\}$	14
5	Hasseovy diagramy pro tři uspořádané množiny	14
6	Hasseův diagram pro (X, R_9)	15
7	Spojový polosvaz, který není svazem	16
8	Průsekový polosvaz, který není svazem	17
9	Svaz	17
10	Popis komunikace aplikace	20
11	Stránka Vlastnosti	26
12	Okno Úprava množiny	28

1 Teoretická část

1.1 Pojem binární relace

V mnoha kontextech se vyskytují vztahy mezi prvky jednotlivých množin. Každý den se setkáváme se vztahy, jako je například vztah mezi podnikem a jeho telefonním číslem, zaměstnancem a jeho platem, nebo například vztahy dané osoby s jeho příbuznými. Vztahy se používají v mnoha oborech matematiky k modelování pojmů jako „je větší než“, „je rovno“ a „dělí“ v aritmetice, „je shodné s“ v geometrii, nebo například „přiléhá k“ v teorii grafů.

Definice 1 (Kartézský součin)

Kartézský součin množin $X_1, \dots, X_n$ je množina $X_1 \times \dots \times X_n$ dána předpisem

$$X_1 \times \dots \times X_n = \{ \langle x_1, \dots, x_n \rangle \mid x_1 \in X_1, \dots, x_n \in X_n \}.$$

Vztahy mezi prvky množin jsou reprezentovány pomocí struktury nazývané **relace**, která je podmnožinou kartézského součinu množin. Nejvhodnějším způsobem jak vyjádřit vztahy mezi prvky dvou množin, je použití uspořádaných dvojic tvořených ze souvisejících prvků. Množinu uspořádaných dvojic nazýváme **binární relací** [1].

Definice 2 (Binární relace na množině)

Binární relací R na množině X rozumíme každou podmnožinu kartézského součinu $X \times X$.

POZNÁMKA 3

Definici bychom mohli vyslovit obecněji pro relaci mezi množinami X a Y , jako podmnožinu kartézského součinu $X \times Y$. Protože bychom tuto definici v následujícím textu nevyužili, omezíme se pouze na relace na množině X .

PŘÍKLAD 4

Na množině $X = \{1, 2, 3, 4\}$ definujeme následující binární relace

- Relace $R_1 = \{ \langle 1, 1 \rangle, \langle 2, 2 \rangle, \langle 3, 3 \rangle, \langle 4, 4 \rangle \}$, v následujícím textu značená symbolem „ $=$ “ (rovná se).
- Relace $R_2 = \{ \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 2 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 3, 3 \rangle, \langle 3, 4 \rangle, \langle 4, 4 \rangle \}$ značená symbolem „ $\leq$ “ (je menší nebo rovno).
- Relace $R_3 = \{ \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 3, 4 \rangle \}$ značená symbolem „ $<$ “ (je ostře menší než).
- Relace $R_4 = \{ \langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 2 \rangle, \langle 2, 4 \rangle, \langle 3, 3 \rangle, \langle 4, 4 \rangle \}$ značená symbolem „ $|$ “ (dělí).

1.2 Vlastnosti binárních relací

Binární relace na množině mají speciální význam, neboť umožňují zkoumat vlastnosti vztahů mezi prvky patřícími do téže množiny.

Definice 5

Nechť X je neprázdná množina a R binární relace na této množině, pak je R

1. **reflexivní**, pokud $\forall x \in X$ platí $\langle x, x \rangle \in R$,
2. **irreflexivní**, pokud $\forall x \in X$ platí $\langle x, x \rangle \notin R$,
3. **symetrická**, pokud $\forall x, y \in X$ platí $\langle x, y \rangle \in R \Rightarrow \langle y, x \rangle \in R$,
4. **asymetrická**, pokud $\forall x, y \in X$ platí $\langle x, y \rangle \in R \Rightarrow \langle y, x \rangle \notin R$,
5. **antisymetrická**, pokud $\forall x, y \in X$ platí $(\langle x, y \rangle \in R \wedge \langle y, x \rangle \in R) \Rightarrow x = y$,
6. **úplná**, pokud $\forall x, y \in X$ platí $\langle x, y \rangle \in R \vee \langle y, x \rangle \in R$,
7. **tranzitivní**, pokud $\forall x, y, z \in X$ platí $(\langle x, y \rangle \in R \wedge \langle y, z \rangle \in R) \Rightarrow \langle x, z \rangle \in R$.

U všech vlastností je velmi důležité, že definují vztah, který musí platit pro všechny prvky nosné množiny, respektive pro všechny dvojice nebo trojice prvků nosné množiny. V žádné z těchto definic pak nesmíme vynechat univerzální kvantifikátory.

V případě, že chceme dokázat, že daná relace některou vlastnost nesplňuje, stačí nalézt jediný prvek, respektive dvojici či trojici prvků, pro kterou daná vlastnost neplatí. Rovněž pokud naopak chceme dokázat platnost dané vlastnosti, pak ji musíme dokázat pro všechny prvky, respektive dvojice či trojice prvků [2].

PŘÍKLAD 6

Vezměme si relaci $R_5 = \{\langle a, a \rangle, \langle a, b \rangle, \langle a, c \rangle, \langle b, b \rangle, \langle b, c \rangle, \langle c, c \rangle\}$ na množině $X = \{a, b, c\}$.

Ověříme, zda je relace R_5 reflexivní, irreflexivní, symetrická, asymetrická, antisymetrická, úplná, tranzitivní.

1. **Reflexivita:** Je třeba ověřit, že každý prvek je v relaci sám se sebou. Je zřejmé, že uvedená vlastnost platí pro každý prvek z množiny X . Relace **je reflexivní**.
2. **Irreflexivita:** Relaci bychom mohli nazvat irreflexivní právě tehdy, pokud by se v relaci nevyskytovaly uspořádané dvojice $\langle a, a \rangle, \langle b, b \rangle, \langle c, c \rangle$. Jestliže je relace reflexivní, pak nemůže být irreflexivní (avšak nemusí mít ani jednu vlastnost z těchto dvou). Relace **není irreflexivní**.

3. **Symetrie:** Je třeba ověřit, že nezáleží na pořadí prvků v relaci, což v našem případě znamená, že pro každou dvojici $\langle x, y \rangle$ platí, že jestliže se nachází dvojice $\langle x, y \rangle$ v relaci, pak i dvojice $\langle y, x \rangle$. Vezměme například dvojici $\langle a, b \rangle$, v relaci se nenachází dvojice $\langle b, a \rangle$, proto již nyní můžeme prohlásit, že daná relace **není symetrická**.
4. **Asymetrie:** Je třeba ověřit, že s každou uspořádanou dvojicí, která je v relaci, opačně uspořádaná dvojice v relaci nebude. Stačí vzít např. uspořádanou dvojici $\langle a, a \rangle$. Tato dvojice v relaci je. Pokud by byla relace asymetrická, pak by tato dvojice musela v relaci zároveň být a nebýt, což nelze. Relace **není asymetrická**.
5. **Antisymetrie:** Je třeba ověřit, že je-li v relaci uspořádaná dvojice $\langle x, y \rangle$ i uspořádaná dvojice $\langle y, x \rangle$, pak nutně musí platit, že $x = y$. Toto tvrzení lze snadno vyvrátit nalezením dvojice, pro kterou pravidlo neplatí. Žádná taková dvojice se v našem příkladu nevyskytuje, tudíž můžeme tvrdit, že relace **je antisymetrická**.
6. **Úplnost:** Tuto vlastnost relace ověříme tak, že se pro každé dva prvky $x, y \in X$ alespoň jedna z dvojic $\langle x, y \rangle, \langle y, x \rangle$ musí nacházet v relaci (úplnost relace implikuje reflexivitu). Relace **je úplná**.
7. **Tranzitivita:** Je třeba ověřit, že pro každou trojici prvků $x, y, z \in X$ platí, že jsou-li v relaci x a y , a současně i y a z , pak nutně musí být v relaci i x a z . Relace **je tranzitivní**.

1.3 Grafické znázornění relací

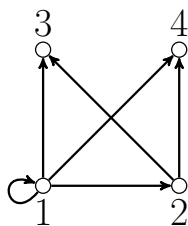
Obecně existuje více způsobů, jak lze binární relace reprezentovat. V této práci se ovšem zaměříme na znázornění **orientovaným grafem a maticí**, které nám poslouží k pochopení dané problematiky. Nutno podotknout, že se v této kapitole zabýváme *pouze znázorněním binárních relací na množině*.

PŘÍKLAD 7

Vezměme si jako příklad relaci R_6 na množině $X = \{1, 2, 3, 4\}$ definovanou vztahem $R_6 = \{\langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle\}$.

1. Orientovaný graf

Nechť R je binární relace na konečné množině $X = \{x_1, x_2, \dots, x_n\}$. Pak můžeme relaci R znázornit orientovaným grafem následujícím způsobem: Pro každý prvek $x_i \in X$ nakreslíme pevný bod a pojmenujeme jej podle x_i . Tento bod nazýváme **vrcholem**. Pokud dvojice $\langle x_i, x_j \rangle \in R$, vykreslíme šipku z x_i do x_j , která se nazývá **orientovaná hrana**. V případě, že se $x_i = x_j$ pak vykreslíme **orientovanou smyčku**. Po vykreslení všech vrcholů a k nim příslušných hran i smyček, získáme **orientovaný graf** [3].



Obrázek 1: Orientovaný graf pro relaci R_6

2. Znázornění maticí

U maticového znázornění relace R sestojíme matici, řekněme M_R , jejíž řádky (a právě tak sloupce) jednoznačně odpovídají prvkům množiny X . V matici M_R bude na řádku odpovídajícím prvku x a ve sloupci odpovídajícím prvku y jednička, pokud $\langle x, y \rangle \in R$, a v opačném případě tam bude nula. Pro výše uvedenou relaci R_6 dostaneme matici

$$M_{R_6} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

v níž řádky odpovídají shora dolů (a sloupce zleva doprava) prvkům 1, 2, 3, 4 [4].

1.4 Relace uspořádání

Často využíváme relace k tomu, abychom uspořádali některé nebo všechny prvky množin. Například číselné množiny jsme zvyklí uspořádat pomocí relace, jenž obsahuje dvojice $\langle x, y \rangle$, kde „číslo x je menší než y “. Pokud přidáme všechny páry ve tvaru $\langle x, x \rangle$, získáme relaci, která je **reflexivní, antisymetrická a tranzitivní**. Jedná se o vlastnosti, které charakterizují relace používané k uspořádání prvků množin.

Definice 8

Nechť R je binární relace na množině X . Potom R nazveme uspořádání, jestliže R je relace reflexivní, antisymetrická a tranzitivní. Je-li R uspořádání na množině X , pak dvojice (X, R) se nazývá uspořádaná množina.

Nachází-li se prvky x, y v relaci R , interpretujeme to slovy „prvek x je menší nebo roven prvku y “. To je v souladu se všemi třemi základními vlastnostmi uspořádání. Uspořádání budeme v dalším textu často značit symbolem „ $\leq$ “ [5].

PŘÍKLAD 9

Relace „je menší nebo rovno“ reprezentovaná symbolem „ $\leq$ “ je uspořádáním na množině celých čísel. Relace je reflexivní, protože platí $x \leq x$ pro každé celé číslo x . Jestliže $x \leq y$ a $y \leq x$, pak nutně $x = y$, tudíž je antisymetrická. A konečně, „ $\leq$ “ je tranzitivní, jelikož $x \leq y$ a $y \leq z$ implikuje vztah $x \leq z$.

PŘÍKLAD 10

Relace „dělitelnosti“ reprezentovaná symbolem „ $|$ “ je uspořádáním na množině přirozených čísel. Relace dělitelnosti definovaná $x | y$ (kde $x, y \in \mathbb{N}$) právě tehdy, když x je dělitel čísla y (tj. když $y = z \cdot x$ pro nějaké $z \in \mathbb{N}$) je relace reflexivní, antisymetrická a tranzitivní. Ano, pro každé přirozené číslo x platí $x | x$, tedy x je dělitelné samo sebou. Rovněž platí, že jestliže $x | y$ a $y | x$, pak $x = y$, a současně jestliže $x | y$ a $y | z$, pak také $x | z$.

Definice 11

Nechť x, y jsou dva prvky uspořádané množiny $(X, \leq)$. Platí-li $x \leq y$ nebo $y \leq x$, jsou prvky x, y porovnatelné, v opačném případě jsou neporovnatelné, což značíme $x \parallel y$.

1.5 Řetězec a antiřetězec

Definice 12

Nechť R je binární relace na množině X , která je reflexivní, antisymetrická, tranzitivní a úplná. Potom R nazveme lineární uspořádání, a dvojici (X, R) lineárně uspořádanou množinou, nebo-li řetězcem.

PŘÍKLAD 13

Relace $R_2 = \{\langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 2 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 3, 3 \rangle, \langle 3, 4 \rangle, \langle 4, 4 \rangle\}$ reprezentovaná symbolem „ $\leq$ “ (je menší nebo rovno) je lineárním uspořádáním na množině $X = \{1, 2, 3, 4\}$.

Definice 14

Je-li dvojice (X, R) uspořádaná množina, pak se každá podmnožina $Y \subseteq X$ nazývá antiřetězec právě tehdy, když jsou každé dva různé prvky Y neporovnatelné v (X, R) .

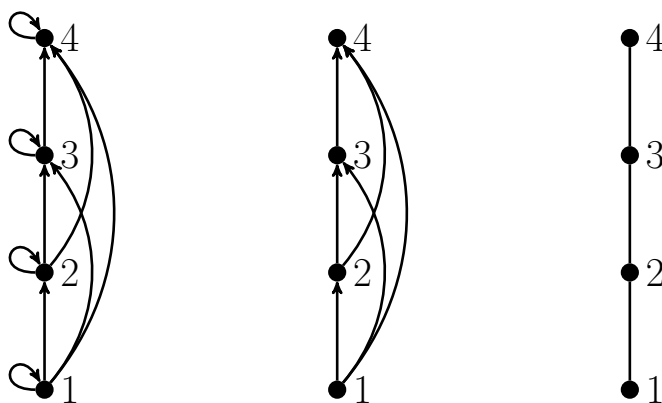
PŘÍKLAD 15

Relace R_1 „rovná se“ reprezentovaná symbolem „ $=$ “ na množině $X = \{1, 2, 3, 4\}$ je antiřetězcem, jelikož nelze porovnat žádné dva různé prvky z množiny X , tj. pro každé dva různé prvky $x, y \in X$ máme $x \parallel y$.

1.6 Relace pokrytí a Hasseovy diagramy

V teorii uspořádání se využívá Hasseův diagram (pojmenovaný po Helmutu Hasseovi) k zobrazení konečné uspořádané množiny. Pro reprezentaci uspořádaných množin využíváme následující postup [6]:

Začneme vykreslením orientovaného grafu relace. Protože je uspořádání reflexivní, existují smyčky $\langle x, x \rangle$ na každém vrcholu x . Tyto smyčky odstraníme. Dále odstraníme všechny hrany $\langle x, y \rangle$, pro které existuje prvek $z \in X$ takový, že $x \leq z$ a $z \leq y$, přičemž prvky x, y, z jsou od sebe navzájem různé. Každou hranu uspořádáme tak, aby její počáteční vrchol byl níže než koncový vrchol. Horizontální umístění vrcholů je vhodné volit tak, aby pokud možno nedocházelo ke křížení hran. Poté odstraníme všechny šipky na cílových hranách, protože nám všechny hrany nyní směřují nahoru ke svému koncovému vrcholu, viz obrázek 2.



Obrázek 2: Postup vytvoření Hasseova diagramu pro (X, R_2)

Definice 16

Nechť dvojice $(X, \leq)$ je uspořádaná množina a „ $\leq$ “ odpovídající relace uspořádání. Prvek x je pokrýván prvkem y , jestliže $x \leq y$ a neexistuje prvek $z \in X$ takový, že $x \leq z$ a $z \leq y$, přičemž prvky x, y, z jsou od sebe navzájem různé. Skutečnost, že prvek x je pokrýván prvkem y budeme vyjadřovat symbolem $x \prec y$. Relaci $\prec$ budeme nazývat relace pokrytí.

Dalším způsobem jak vykreslit Hasseův diagram je využít příslušnou relaci pokrytí. Prvky množiny znázorníme jako body, kde v případě, že $x \leq y$, pak bod x bude znázorněn níže než bod y . Hranu mezi prvky x, y vykreslíme právě tehdy, když $x \prec y$ [2].

PŘÍKLAD 17

Příslušná relace pokrytí $R_{\prec} = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$ pro uspořádanou množinu (X, R_2) , jenž je znázorněna na obrázku 2.

1.7 Speciální prvky uspořádaných množin

Představme si nyní některé speciální prvky v uspořádaných množinách a jejich vzájemné vztahy.

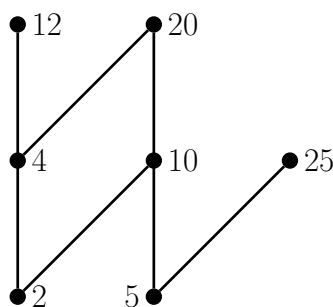
Definice 18

Nechť dvojice $(X, \leq)$ je uspořádaná množina a „ $\leq$ “ odpovídající relace uspořádání. Prvek $x \in X$ je:

1. **nejmenší prvek** X právě tehdy, když $\forall y \in X$ platí $x \leq y$,
2. **největší prvek** X právě tehdy, když $\forall y \in X$ platí $y \leq x$,
3. **minimální prvek** X právě tehdy, když $\forall y \in X$ platí $y \leq x \Rightarrow y = x$ (tedy když žádný prvek y množiny X není menší než prvek x),
4. **maximální prvek** X právě tehdy, když $\forall y \in X$ platí $x \leq y \Rightarrow x = y$ (tedy když žádný prvek y množiny X není větší než prvek x).

Maximální prvek obecně nemusí být největším prvkem, tudíž zatímco není menší než libovolný jiný prvek, nemusí být větší než všechny ostatní prvky (totéž platí o minimálním prvkem, který obecně nemusí být nejmenším prvkem). Maximálních prvků může být v jedné množině víc. Pokud však množina má největší (nejmenší) prvek, je tento prvek jediným maximálním (minimálním) prvkem [5].

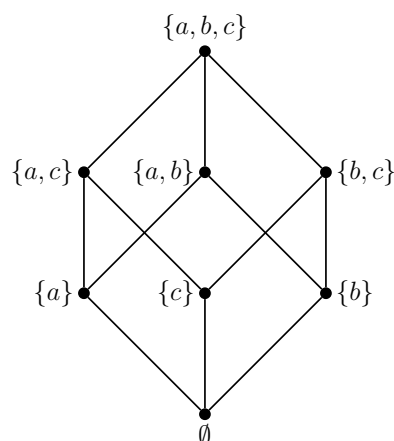
Tvrzení nyní podložíme příkladem pro uspořádanou množinu (X, R_7) , kde $X = \{2, 4, 5, 10, 12, 20, 25\}$ a $R_7 = \{\langle 2, 2 \rangle, \langle 2, 4 \rangle, \langle 2, 10 \rangle, \langle 2, 12 \rangle, \langle 2, 20 \rangle, \langle 4, 4 \rangle, \langle 4, 12 \rangle, \langle 4, 20 \rangle, \langle 5, 5 \rangle, \langle 5, 10 \rangle, \langle 5, 20 \rangle, \langle 5, 25 \rangle, \langle 10, 10 \rangle, \langle 10, 20 \rangle, \langle 12, 12 \rangle, \langle 20, 20 \rangle, \langle 25, 25 \rangle\}$. Relace R_7 odpovídá relaci dělitelnosti na X .



Obrázek 3: Hasseův diagram pro (X, R_7)

PŘÍKLAD 19

Hasseův diagram na obrázku 3 pro (X, R_7) ukazuje, že maximální jsou prvky 12, 20 a 25 a minimální jsou prvky 2 a 5. Nejmenší ani největší prvek neexistuje.

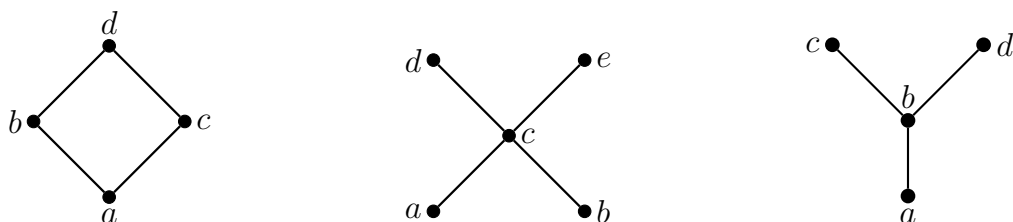


Obrázek 4: Hasseův diagram pro $(P(X), \subseteq)$, kde $X = \{a, b, c\}$

PŘÍKLAD 20

Mějme uspořádanou množinu $(P(X), \subseteq)$ definovanou Hasseovým diagramem na obrázku 4, tedy potenční množinu tříprvkové množiny $X = \{a, b, c\}$ uspořádanou relací „býti podmnožinou“. Nejmenším prvkem je prázdná množina, největším prvkem pak samotná množina X .

Lze si povšimnout, že každá uspořádaná množina z příkladů má nejvýše jeden nejmenší a jeden největší prvek. Že tomu tak vždy bude, můžeme snadno nahlédnout přímo z definice nejmenšího a největšího prvku. Pokud by byly dva prvky x, y uspořádané množiny nejmenší, pak by muselo platit, že $x \leq y$ a $y \leq x$. Protože víme, že uspořádání je antisymetrická relace, musí $x = y$. To znamená, že takový prvek je pouze jeden [2].



Obrázek 5: Hasseovy diagramy pro tři uspořádané množiny

PŘÍKLAD 21

Vezměme si postupně zleva doprava jednotlivé uspořádané množiny z obrázku 5. U první je nejmenším prvkem a a největším d . Ve druhé jsou minimální prvky a, b , maximální d, e . Největší ani nejmenší prvek neexistuje. Třetí uspořádaná množina má nejmenší prvek a , dva maximální prvky c, d , přičemž největší prvek neexistuje.

POZNÁMKA 22

Nechť dvojice $(X, \leq)$ je uspořádaná množina a „ $\leq$ “ odpovídající relace uspořádání. Pak platí:

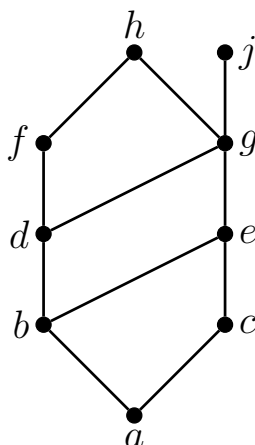
1. Je-li $x \in X$ nejmenší prvek, pak je také minimální prvek.
2. Je-li $x \in X$ největší prvek, pak je také maximální prvek.

1.8 Dolní a horní kužel, supremum a infimum

Definice 23

Nechť dvojice $(X, \leq)$ je uspořádaná množina a $Y \subseteq X$. Označme $H(Y) = \{x \in X; y \leq x \text{ pro každé } y \in Y\}$,
 $D(Y) = \{x \in X; x \leq y \text{ pro každé } y \in Y\}$.
Pak se množina $H(Y)$ se nazývá horní kužel množiny Y a množina $D(Y)$ se nazývá dolní kužel množiny Y .

Z definice nám vyplývá, že horní kužel množiny Y v uspořádané množině $(X, \leq)$ obsahuje prvky z X , které jsou větší nebo rovny všem prvkům obsaženým v Y (analogicky pro dolní kužel) [6].



Obrázek 6: Hasseův diagram pro (X, R_9)

PŘÍKLAD 24

Určeme dolní a horní kužel pro množiny $Y_1 = \{a, b, c\}$ a $Y_2 = \{a, c, d, f\}$ v uspořádané množině, která je definována Hasseovým diagramem na obrázku 6. Pro množinu Y_1 je $D(Y_1) = \{a\}$ a $H(Y_1) = \{e, g, h, j\}$. Pro množinu Y_2 pak je $D(Y_2) = \{a\}$ a $H(Y_2) = \{h\}$.

Definice 25

Má-li $H(Y)$ nejmenší prvek, pak se tento prvek nazývá *supremum* Y a značí se $\sup(Y)$. Analogicky má-li $D(Y)$ největší prvek, pak se tento prvek nazývá *infimum* Y a značí se $\inf(Y)$.

PŘÍKLAD 26

Určeme supremum a infimum u podmnožiny $Y = \{3, 9, 12\}$, pokud existují na uspořádané množině $(\mathbb{N}, |)$

Do dolního kužele podmnožiny Y patří každé přirozené číslo takové, že všechny prvky této podmnožiny jsou tímto číslem dělitelné. Taková čísla jsou pouze dvě, 1 a 3. Protože však $1 \mid 3$, je 3 infimem $D(Y)$. Podobně jako u infima nyní nalezneme supremum. Do horního kužele podmnožiny Y patří každé přirozené číslo, právě když je dělitelné čísly 3, 9 a 12. Přirozená čísla s touto vlastností jsou ty, která jsou dělitelná nejmenším společným násobkem 3, 9 a 12, což je 36. Proto je 36 supremem $H(Y)$.

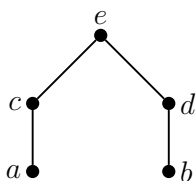
Infimum množiny $Y \subseteq X$ neexistuje pokud $D(Y)$ neobsahuje největší prvek. Největší prvek obecně existovat nemusí, takže ani infimum obecně existovat nemusí (analogicky pro supremum). Pokud však infimum případně supremum či obojí existuje, pak je určeno jednoznačně [2].

1.9 Svazy a polosvazy

V teorii množin se vyskytují takové uspořádané množiny, u kterých existuje pro každou dvojici prvků infimum, supremum, nebo obojí.

Definice 27

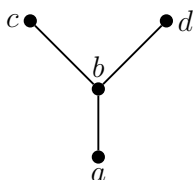
Nechť dvojice $(X, \leq)$ je uspořádaná množina, v níž ke každým dvěma prvkům existuje supremum, pak se daná uspořádaná množina nazývá *spojový polosvaz*.



Obrázek 7: Spojový polosvaz, který není svazem

Definice 28

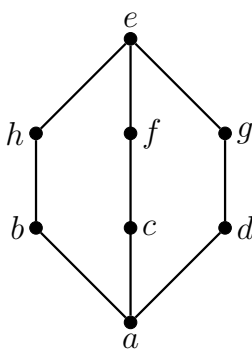
Nechť dvojice $(X, \leq)$ je uspořádaná množina, v níž ke každým dvěma prvkům existuje infimum, pak se daná uspořádaná množina nazývá průsekový polosvaz.



Obrázek 8: Průsekový polosvaz, který není svazem

Definice 29

Nechť dvojice $(X, \leq)$ je uspořádaná množina, v níž ke každým dvěma prvkům existuje supremum i infimum, pak se daná uspořádaná množina nazývá svaz.



Obrázek 9: Svaz

Každá lineárně uspořádaná množina je svazem, jelikož jsou každé dva prvky porovnatelné, máme buď $x \leq y$ nebo $y \leq x$ pro libovolné dva prvky $x, y \in X$. Pokud lze každé dva prvky porovnat, lze pro ně nalézt infimum i supremum. Číselné množiny $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ a $\mathbb{R}$ uspořádané relací „menší nebo rovno“ jsou svazy, ve kterých infimum a supremum odpovídá minimu a maximu z dané dvojice prvků [6].

Dalším příkladem je množina přirozených čísel, uspořádaná dělitelností, kde pro každé dva prvky je supremum nejmenší společný násobek a infimum je největší společný dělitel.

2 Implementace

2.1 Použité technologie

K vytvoření aplikace bylo použito vývojové prostředí **Microsoft Visual Studio 2015** na platformě .NET Framework 4.5 s využitím grafického subsystému Windows Presentation Foundation (WPF). Aplikace byla vytvořena a testována pro Microsoft Windows 8 a novější.

2.2 WPF

Zkratka **WPF** (Windows Presentation Foundation) představuje grafický Framework pro psaní aplikací od společnosti Microsoft. WPF nám umožňuje vytvářet aplikace s pokročilou grafickou úrovní, které by se například ve Windows Forms zvládaly jen obtížně. Pro psaní WPF aplikací se využívá značkovací jazyk **XAML** spolu s **.NET jazykem** (C#, Visual Basic). Tyto aplikace se renderují na grafické kartě a tím snižují zatížení CPU při vytváření graficky náročných aplikací.

Extensible Application Markup Language (XAML) je značkovací jazyk (obdoba HTML), založený na XML a využívaný k popisu grafického rozhraní v aplikacích společnosti Microsoft nové generace. Zkratka původně znamenala Extensible Avalon Markup Language, kde Avalon bylo kódové označení pro Windows Presentation Foundation (WPF). Je určený primárně k návrhu uživatelského prostředí pro technologie WPF a Silverlight [7].

2.3 Použité knihovny

K zobrazení výukové textu v aplikaci byla využita knihovna **MoonPdf**. MoonPdf je PDF prohlížeč vytvořený pro WPF aplikaci, který využívá knihovnu MoonPdfLib. Tato knihovna obsahuje ovládací prvky, využívané v mé aplikaci.

Více na www.codeproject.com.

2.4 Reprezentace dat

- **Mypoint** – pro reprezentaci bodu v grafické podobě se využívá třída *MyPoint.cs*. Obsahuje objekt typu *Point*, který značí středové souřadnice vykresleného bodu. Grafická podoba bodu se skládá z *Ellipse* a *String*. Při vytváření bodu se díky parametrům, které se předávají v konstruktoru získá středový bod, text který bude zobrazen, barva rámečku a textu.
- **Reprezentace párů** – pro reprezentaci párů se využívá třída *Pair*. Obsahující dvě proměnné *First* a *Second* reprezentující jednotlivé prvky páru (jednotlivé prvky párů mohou být libovolné datové typy).

```
Pair<string, string> pair = new Pair<string, string>("first element", "second element");
```
- **Reprezentace relací** – relace se skládají z listů typu *Pair*.

```
List<Pair<string, string>> relation = new List<Pair<string, string>>();
```


Přidání/odstranění/editace prvků v relaci probíhá pomocí metod *Listu* (implementováno od Microsoftu), například *relation.Add()*, *relation.Remove()*. Pro úpravu lze využít více postupů, například *relace[index].First = ""*;

```
relace[index].First = "";
```
- **Reprezentace množiny** – množina je reprezentována pomocí listu datového typu, který využívají prvky v páru. Pokud se pár skládá ze *stringů* => pak i množina je list typu *string*.

```
List<string> Set = new List<string>();
```


Přidání/odstranění/editace prvků probíhá stejným způsobem jako u relací.

2.5 MVVM

Jedná se o návrhový vzor aplikace, usnadňující práci programátora tím, že odděluje logiku od grafického rozhraní aplikace. Kód je poté lépe udržitelný, usnadňuje testování a případný další vývoj aplikace. Mezi jeho základní vrstvy patří [8]:

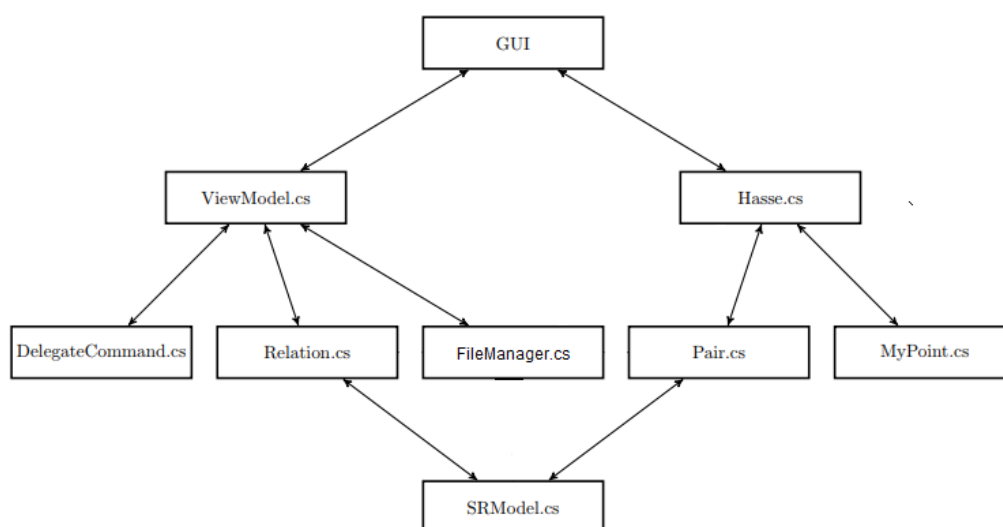
- **Model** – popisuje datovou strukturu jednotlivých objektů, které jsou využívány k zobrazení/ukládání informací od uživatele.
- **View** – představuje GUI aplikace, které slouží pro interakci s uživatelem.
- **ViewModel** – propojuje dvě výše zmíněné vrstvy v celek. Zpracovává veškeré požadavky uživatele a z toho plynoucí operace s daty. Pro usnadnění práce vznikly nové „operace“ *Binding* a *Command*.
 - *Binding* slouží k nalinkování vybraných datových struktur do jednotlivých grafických elementů, ať už jen pro čtení dat případně přímé

propsání vstupu do datových struktur. Dále také slouží k namapování jednotlivých commandů na tlačítka, případně další ovládací prvky.

V případě bindingování dat je nutné implementovat `INotifyCollectionChanged` a `INotifyPropertyChanged` která nám zajišťují vyslání událostí při změně jednotlivých dat a následné aktualizování GUI elementu.

- *Command* slouží k vyvolání nějaké akce po uživatelské interakci na vybraný grafický element, ke kterému je nabindingovaný .

2.6 Struktura aplikace



Obrázek 10: Popis komunikace aplikace

2.7 Technika kreslení Hasseových diagramů a jejich využití ve výpočtech

Modely: `SRModel`, `MyPoint`, `Pair`.

Při vytváření objektu předáváme **Relation**, obsahující funkce a modely pro množinu a relaci. Je nutné vytvořit **RetList** typu `<string, int>`, který si uchovává jednotlivé prvky z množiny a jejich úroveň (tedy „y souřadnici“), na které bude prvek umístěn v grafu.

Dále si vytvoříme **PointLst** `<MyPoint>`, který obsahuje grafickou podobu jednotlivých prvků z množiny. Vytvoření Hasseova diagramu probíhá v metodě `LoadData()`:

1. Do listu `RetList` se uloží prvky z množiny s počtem jejich výskytů v relaci na levé straně jednotlivých párů. Každý pár se poté skládá z daného prvku množiny (na pozici *first*) a počtu jeho vazeb (na pozici *second*).

2. RetList se seřadí podle výskytů a zavolá se metoda *SetZero*, která počet výskytů vynuluje. Tento krok považujeme za velmi podstatný, jelikož nám setřídí prvky tak, aby splňovali vlastnost tranzitivity. Následně je tato proměnná využívána pro ukládání úrovně prvku v Hasseových diagramech.
3. Na RetList se následně zavolá metoda *Level*, kde tento list procházíme for cykly (pomocí něj nám vzniknout všechny kombinace párů). Jestliže se tento pár nachází v relaci, zvýší se úroveň prvku second z daného páru. Po celkové kontrole RetList obsahuje prvky z množiny a jejich úroveň.
4. Metoda *GetPoints* pomocí RetListu vytváří body ukládané v PointList. RetList se postupně prochází a vybírají se prvky na stejné úrovni. Dle vzorčeků se vytvoří bod pro každý prvek z RetListu a přiřadí se mu souřadnice x a y . Tyto body se ukládají do PointList.
5. Metoda *Drawlines* vykresluje hrany mezi jednotlivými body v závislosti na jednoduchém vzorci: Dva vrcholy se spojí hranou vedenou od prvku x k prvku y , jestliže $x \leq y$ a neexistuje z takové, že $x \leq z \leq y$. Odtud si uchováváme body, které spojíme hranou a vypisujeme relaci pokrytí.

2.8 Výpočty založené na Hasseových diagramech

Porovnávání počtu vazeb, následné vynulování a nastavení úrovní nám slouží v programu pro výpočty následujících pojmů: minimální, maximální, nejmenší a největší prvky uspořádaných množin, horní a dolní kužel, infimum a supremum, svazy a polosvazy.

Vezměme si například minimální, maximální, největší a nejmenší prvky. Uspořádáme zadaný vstup podle počtu vazeb (od největšího po nejmenší), poté zleva doprava zkontrolujeme, zda mezi zadanými prvky existuje vazba. Pokud vazba mezi prvky existuje (*IsPairInRelation*), úroveň prvku, se kterým daný prvek porovnáváme se zvýší. Při nastavování úrovní nám pak na úrovni 0 zůstanou pouze prvky dle definic.

Pak už nám jen stačí ověřit, zda-li minimální prvek není pouze jeden. Dle definice víme, že pokud ano, pak je tento prvek i nejmenší. Podobně pokud chceme získat prvky maximální, list prvků, který obsahuje počet vazeb jednoduše obrátíme pomocí metody *Reverse()*.

2.9 Popis tříd

Třída `SRModel.cs` představuje reprezentaci `Set`, `SubSet` a `Relation` pomocí listů a další pomocné struktury.

Třída `Relation.cs` představuje fukcionalitu pro třídu `SRModel.cs`, následně využitá ve třídě `ViewModel.cs`.

- `public void CheckSubSet()` – ověří, zda podmnožina obsahuje prvek, pokud ne, přidá jej.
- `public bool Reflexive()` – ověří vlastnost relace a vrátí hodnoty `true/false`, pokud je/není relace reflexivní.
- `public bool Antisymmetric()` – ověří vlastnost relace a vrátí hodnoty `true/false`, pokud je/není relace antisymetrická.
- `public bool Irreflexive()` – ověří vlastnost relace a vrátí hodnoty `true/false`, pokud je/není relace irreflexivní.
- `public bool Asymmetric()` – ověří vlastnost relace a vrátí hodnoty `true/false`, pokud je/není relace asymetrická.
- `public bool Transitive()` – ověří vlastnost relace a vrátí hodnoty `true/false`, pokud je/není relace tranzitivní.
- `public bool IsPairInRelation(Pair<string, string> p)` – ověří, zda se hledaný pár nachází v relaci.
- `public void ReflexUzaver()` – přidá dvojice pro splnění vlastnosti reflexivity.
- `private bool TranzAddPair()` – přidá dvojice pro splnění vlastnosti tranzitivity.
- `public void GenerateRetezec()` – generuje řetězec na zadané množině.
- `public void GenerateAntiretezec()` – generuje antiřetězec na zadané množině.
- `public void GenerateRelation()` – generuje uspořádání na zadané množině.
- `public void GenerateSLattice()` – generuje spojový polosvaz na zadané množině.
- `public void GeneratePLattice()` – generuje průsekový polosvaz na zadané množině.

- `public void GenerateLattice()` – generuje svaz na zadané množině.
- `public List<string> Maximal()` – vrací list maximálních prvků v relaci.
- `public List<string> Minimal()` – vrací list minimálních prvků v relaci.
- `public bool Antiretezec()` – ověří, zda je relace složená z párů stejných prvků, vrací true/false pokud ano/ne.
- `private setZero(List<Pair<string, int>> lst)` – vynuluje počet vazeb daného prvku na pozici second.
- `private void Level(List<Pair<string, int>> lst)` – ověří vlastnost relace a vrací úroveň prvků.
- `public List<Pair<string, int>> UpperBound()` – vrací list prvků, které jsou horním kuželem pro vloženou podmnožinu.
- `public string Supremum(List<Pair<string, int>> value)` – metoda nalezne minimální prvek horního kužele, pokud jich nalezne více, pak vrací, že neexistuje.
- `public List<Pair<string, int>> LowerBound()` – vrací list prvků, které jsou dolním kuželem pro vloženou podmnožinu.
- `public string Infimum(List<Pair<string, int>> value)` – metoda nalezne maximální prvek dolního kužele, pokud jich nalezne více, pak vrací, že neexistuje.
- `public bool Total()` – ověří vlastnost relace a vrací hodnoty true/false, pokud je/není relace úplná.
- `public bool PLattice()` – prověří uspořádanou množinu a vrací hodnoty true/false, pokud je/není průsekový polosvazem.
- `public bool SLattice()` – prověří uspořádanou množinu a vrací hodnoty true/false, pokud je/není spojový polosvazem.

Třída `ViewModel.cs` využívá funkcionalitu třídy `Relation.cs`, výstupy z metod zpracovává ve svých vlastních a předává je grafickému prostředí. Slouží také k získání uživatelských vstupů, ať už se jedná o data nebo události, které vyvolá uživatel kliknutím na tlačítka.

Třída `MyPoint.cs` – datový model pro grafické znázornění bodu.

- `public MyPoint(Point p, string name, Brush colorText, Brush colorFrame)`
 - `Point p` – bod se středovými souřadnicemi.
 - `string name` – název bodu, který se vykreslí.
 - `brush colorText/Frame` – barva textu/rámečku.
- `public Polyline MakeLine(MyPoint p)` – vrací grafickou podobu úsečky mezi aktuálním a předaným bodem.
- `public void Run()` – vytvoří grafickou podobu bodu ze získaných dat.

Třída `Pair.cs` – datový model pro reprezentaci páru.

- `public Pair(T first, U second)` – vytvoří pár se získanými parametry.
- `public override string ToString()` – vrací podobu páru.

Třída `DelegateCommand.cs` implementuje rozhraní *ICommand*. Jedná se o metody, které představují náhradu za GUI řízené pomocí událostí.

Třída `FileManager.cs` slouží pro ukládání zadání, obsahuje interní třídu `FileManagerData`, která slouží k serializaci a deserializaci při zápisu/čtení dat.

- `public static void WriteToFile(string data)` – slouží pro uložení dat do souboru.
- `public static string ReadFromFile()` – slouží pro načtení dat ze souboru.

Třída `Hasse.cs` popsána v kapitole 2.7.

3 Uživatelská příručka

3.1 Požadavky

Aplikace byla testována a vyvíjena pro operační systém Windows 8 a vyšší. Minimální rozlišení je 1000 x 740 pixelů. Maximální rozlišení bylo nastaveno na 1920 x 1200 pixelů.

3.2 Instalace aplikace

Aplikace „Uspořádané množiny“ se nainstaluje spuštěním souboru „setup.exe“ umístěném v adresáři bin, který se nachází na přiloženém DVD. Po zobrazení okna instalátoru je nutné pokračovat kliknutím na tlačítko „Next“. V dalším okně vyberte adresář pro umístění aplikace a opět klikněte na tlačítko „Next“. Poté potvrďte poslední souhrnné okno, čímž se zahájí instalace.

3.3 Aplikace

Aplikace „Uspořádané množiny“ se skládá ze dvou částí.

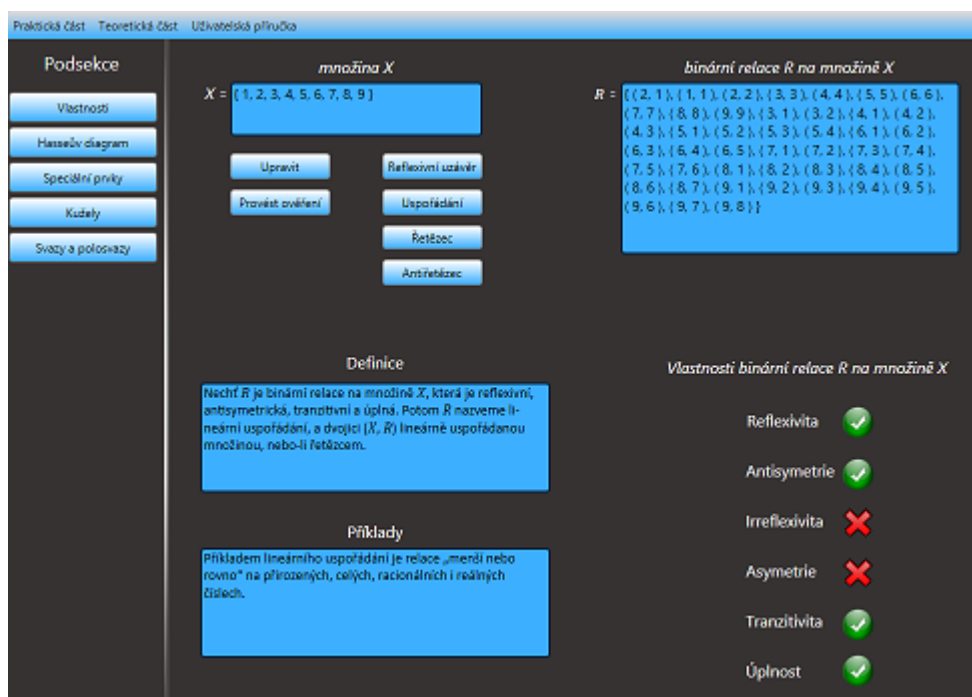
- **Stránkovací menu**, které se nachází v horní části po celé šířce aplikace. Toto menu nám slouží pro přepínání obsahu.
- **Stránky**, které se nám otevírají v závislosti na volbě v menu.

3.3.1 Praktická část

Při kliknutí na položku menu *Praktická část* se nám otevře stránka obsahující v levé části po celé výšce submenu vytvořené z tlačítek. Jednotlivá tlačítka slouží k otevření podstránek, kde dochází k výpočtům spojenými s uspořádanými množinami, viz obrázek 11.

Tlačítka sloužící pro přepnutí obsahu:

- *Vlastnosti*
- *Hasseovy diagramy*
- *Speciální prvky*
- *Kužely*
- *Svazy a polosvazy*



Obrázek 11: Stránka Vlastnosti

3.3.1.1 Stránka Vlastnosti

Stránka *Vlastnosti* slouží pro ověření vlastností binární relace na množině.

V horní části stránky se nachází textové pole pro množinu a relaci. Upravit zadání lze pomocí tlačítek *Upravit* a *Generovat*, které nám umožní vygenerovat uspořádání, řetězec nebo antiřetězec na již zadané množině. Po změně zadání je nutné spustit výpočet pomocí tlačítka *Ověřit*. V případě, že zadání změním na jedné podstránce, se výpočet vždy provede automaticky i pro ostatní podstránky. Pokud je zadání nevyhovující (špatně zadaný vstup, případně není binární relace na množině uspořádáním), tlačítka v submenu zešednou a nelze provést výpočet, kromě ověření vlastností.

V dolní části v levém rohu nalezneme dvě další textová pole provádějící výpis v závislosti na vlastnostech, které daná relace splňuje. V pravém rohu nalezneme grafické ověření vlastností pro větší přehlednost viz obrázek 11.

3.3.1.2 Stránka Hasseovy diagramy

Stránka *Hasseovy diagramy* představuje Hasseův diagram spolu s informacemi o daném zobrazení.

V horní části nalezneme textové pole pro množinu a relaci. V dolní části v levém rohu nalezneme plátno, v němž se daný Hasseův diagram vykreslí. V pravém rohu pak opět nalezneme dodatečné informace k danému Hasseovu diagramu.

3.3.1.3 Stránka Speciální prvky

Stránka *Speciální prvky* ověří, které prvky v daném uspořádání označujeme jako nejmenší, největší, maximální či minimální.

V horní části nalezneme textové pole pro množinu a relaci. V dolní části v levém rohu nalezneme plátno, v němž se daný Hasseův diagram vykreslí. Speciální prvky jsou zde barevně značeny. Minimální/maximální prvky jsou zde zvýrazněny zeleným/červeným rámečkem. Podobně pokud je prvek nejmenší/největší, pak je jeho text zbarven zeleně/červeně. Pokud existuje prvek, který je minimálním i maximálním/nejmenším i největším současně (dle definice), pak je tento prvek vyznačen zlatým rámečkem/textem. V pravém rohu pak opět nalezneme dodatečné informace k danému Hasseovu diagramu.

3.3.1.4 Stránka Kužely

Stránka *Kužely* předvede pro podmnožinu Y množiny X horní/dolní kužel, supremum a infimum.

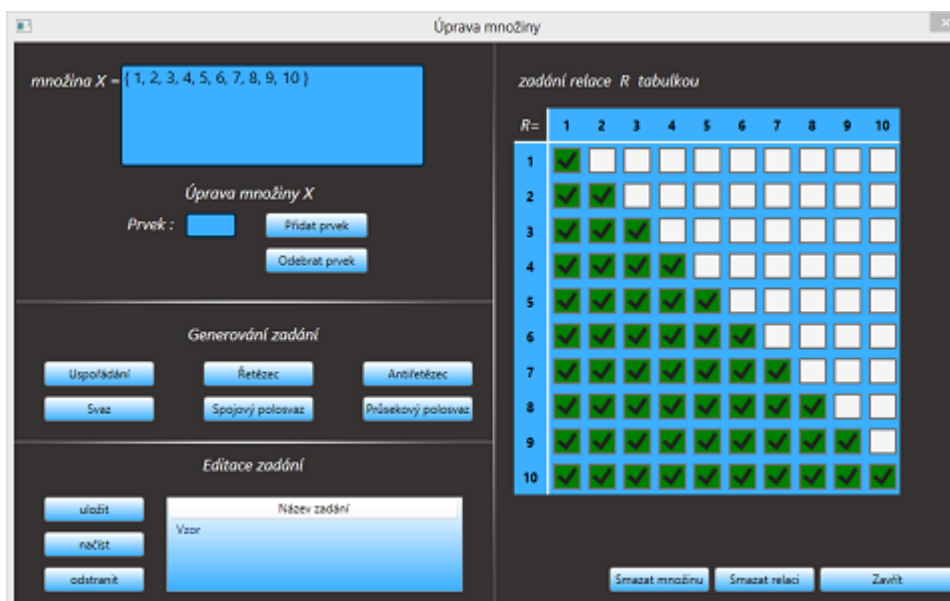
V horní části nalezneme podobně jako u stránky *Vlastnosti* textové pole pro množinu a relaci. Avšak, protože zde nyní využíváme i podmnožinu Y , je horní část rozšířena o textové pole pro tuto podmnožinu, do které lze vkládat prvky. Prvky lze do podmnožiny Y vkládat dvojím způsobem, při zapsání prvku do textboxu můžeme kliknout na tlačítko *Přidat prvek*, případně zmáčknout klávesu *Enter*. Pro odmazání prvku z množiny lze využít tlačítko *Smazat prvek* nebo klávesu *Escape*. U podmnožiny lze vkládat pouze prvky, které se vyskytují v množině.

V dolní části v levém rohu nalezneme plátno, v němž se daný Hasseův diagram vykreslí. Kužely jsou zde barevně značeny. Horní/dolní kužel je zde zvýrazněn zeleným/červeným rámečkem. Podobně pokud je prvek supremem/infimem, pak je jeho text zbarven zeleně/červeně. V případě, že je prvek současně infimem i supremem/horním i dolním kuželem, je označen zlatou barvou/zlatým rámečkem. V pravém rohu pak nalezneme textová pole obsahující informace o nalezených prvcích.

3.3.1.5 Stránka Svazy a polosvazy

Stránka *Svazy a polosvazy* ověří, zda lze danou uspořádanou množinu nazvat průsekovým/spojovým polosvazem, případně svazem.

V horní části nalezneme textové pole pro množinu a relaci. V dolní části nalezneme přehled hodnot, přičemž můžeme volit mezi zobrazením v tabulce, nebo v listech. List slouží pro výsledky, kde jedno textové pole slouží pro spočítané dvojice a druhé pro dvojice, kde supremum/infimum neexistuje. Obdobně tabulka zobrazí znak „-“ v případě, že supremum/infimum neexistuje. V dolní části v pravém rohu nalezneme grafické ověření, zda lze zadanou uspořádanou množinu nazvat svazem, případně polosvazem.



Obrázek 12: Okno Úprava množiny

3.3.1.6 Okno Úprava množiny

Na výše zmíněných stránkách se vyskytuje tlačítko *Upravit*, které umožňuje změnit zadání pro výpočty. Při kliknutí se otevře okno *Úprava množiny*, které je rozdělené do několika částí.

V levém horním rohu lze upravit množinu X , prvky do množiny přidáme zapsáním do textového pole a následným kliknutím na tlačítko *Přidat prvek*. Přidat prvek lze i využitím klávesy *Enter*. Obdobně lze odstranit prvky, kde využíváme klávesy *Escape*, případně tlačítko *Odebrat prvek*.

V pravé části lze využít tabulku pro úpravu relace, která se vytváří v závislosti na zadané množině X . Hodnota ve sloupci představuje první prvek v páru, řádek představuje druhý prvek. Dvojici (neboli pár) následně přidáme zakliknutím okénka, které odpovídá námi zvoleným hodnotám. Odebrat dvojici lze opětovným zakliknutím příslušného okna. Pod tabulkou nalezneme tlačítka pro odstranění hodnot z množiny, případně z relace a tlačítko pro zavření okna.

Další část nám slouží pro generování zadání, nalezneme ji hned pod úpravou množiny X . Zadání se vždy generuje na námi zadané množině.

V případě, že si chceme některé ze zadání uložit pro pozdější práci, lze využít *File manager* v levém dolním rohu. Při kliknutí na tlačítko uložit vyskočí okénko, do kterého si lze uložit zadání pod vlastním názvem. V případě, že zaklikneme jedno ze zadání, lze toto zadání načíst, případně odstranit.

3.3.2 Teoretická část

Při kliknutí na položku menu *Teoretická část* se nám otevře stránka obsahující v horní části po celé šířce menu vytvořené z tlačítek a seznamů. V dolní části dochází k zobrazení pdf dokumentu.

- Menu *Kapitoly*, které po rozkliknutí otevře nabídku kapitol.
 - Binární relace
 - Vlastnosti binárních relací
 - Relace uspořádání
 - Řetězec a antiřetězec
 - Relace pokrytí a Hasseovy diagramy
 - Speciální prvky uspořádaných množin
 - Horní a dolní kužel, supremum a infimum
 - Svazy a polosvazy
- Menu *Náhled*
 - Kniha
 - Jednotlivě
- Menu *Velikost*
 - Přiblížit
 - Oddálit
- Tlačítko *Přejít na stranu*, které nám otevírá okno Navigace.

3.4 Odinstalování aplikace

Odinstalování aplikace se provádí standardním způsobem v nabídce Start→Ovládací panely→Odinstalovat program vybráním položky Uspořádané množiny a kliknutím na tlačítko *Odinstalovat*.

Závěr

Byla vytvořena aplikace na podporu výuky uspořádaných množin, která provádí výpočty na předem zadaném a ověřeném vstupu. Aplikace slouží zejména jako pomůcka pro studenty a zaměřuje se na základní pojmy uspořádaných množin a binárních relací na množině. Součástí práce je výukový text, jenž poslouží především studentům, kteří se s těmito pojmy seznamují poprvé.

Při návrhu uživatelského prostředí byl kladen důraz především na jednoduchost a názornost. Rozšířit aplikaci lze například implementací ošetřující křížení hran u Hasseových diagramů a rozšíření obsahu o další pojmy teorie množin.

Conclusions

The aim of the bachelor's thesis was to create an application for support of teaching partially ordered sets. This application is mainly for students and aims at basic terms of ordered sets and binary relations on a sets. The part of this study is educational text helping students who are acquainted to this terms for the first time.

The aim of creating user's interface was for it to be simple and objective. Extension of this application is possible for example by implementation regulates crossing edges of Hasse's diagrams and extending of content by another terms of set's theory.

4 Obsah přiloženého DVD

Součástí této práce je DVD. Jeho obsahem jsou tři adresáře a jeden soubor.

bin/

Instalační balíček `SETUP.EXE` vytvořené aplikace.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh.

src/

Kompletní zdrojové texty programu `USPOŘÁDANÉ MNOŽINY`.

readme.txt

Instrukce pro instalaci a spuštění programu `USPOŘÁDANÉ MNOŽINY`.

Bibliografie

- [1] Bělohlávek Radim, Vychodil Vilém: *Diskrétní matematika pro informatiky I*. Olomouc, 2006.
- [2] Bělohlávek Radim, Vychodil Vilém: *Diskrétní matematika pro informatiky II*. Olomouc, 2006.
- [3] Wikipedie, otevřená encyklopedie, www.wikipedia.org/orientovany.graf.
- [4] Wikipedie, otevřená encyklopedie, www.wikipedia.org/matice.
- [5] Beran Ladislav: *Uspořádané množiny*. Mladá fronta, Praha, 1978.
- [6] Rosen Kenneth: *Discrete Mathematics and Its Applications (2nd ed.)*. McGraw-Hill, New York, USA, 1991.
- [7] Petzold Charles: *Mistrovství ve Windows Presentation Foundation*. Computer Press, a.s., Brno, 2008.
- [8] Wikipedie, otevřená encyklopedie, www.wikipedia.org/mvvm.