VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

# VOLNÉ ALGEBRAICKÉ STRUKTURY A JEJICH VYUŽITÍ PRO SEGMENTACI DIGITÁLNÍHO OBRAZU

FREE ALGEBRAIC STRUCTURES AND THEIR APPLICATION FOR SEGMENTATION OF A DIGITAL IMAGE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. KATEŘINA ČAMBALOVÁ

VEDOUCÍ PRÁCE
SUPERVISOR

Mgr. JAN PAVLÍK, Ph.D.

BRNO 2015

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav matematiky
Akademický rok: 2014/2015

# ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Kateřina Čambalová

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Matematické inženýrství (3901T021)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

**Volné algebraické struktury a jejich využití pro segmentaci digitálního obrazu**

v anglickém jazyce:

**Free algebraic structures and their application for segmentation of a digital image**

Stručná charakteristika problematiky úkolu:

Práce zasahuje do algebry a počítačové grafiky a spočívá v algebraickém popisu metody pro segmentaci digitálního obrazu a jejím zobecnění. Cílem je studium základů univerzální algebry a některých metod v analýze obrazu, zejména těch, které využívající její prostředky. Např. metoda svazové fuzzy segmentace využívá volné distributivní svazy, přičemž bohatost algebraické struktury zde odpovídá šíři možností metody. V tomto případě by se tedy jednalo např. o možné zobecnění metody využívající Booleovy svazy, což by dávalo širší možnosti aplikace.

Cíle diplomové práce:

Jedná se o základní výzkum, hlavní část práce je v pochopení a přehledném zpracování příslušných odvětví univerzální algebry a analýzy obrazu. Teoretické odvození nové rozšířené metody.

Seznam odborné literatury:

Gabor T. Herman: Geometry of Digital Spaces, Springer 1998

J. Pavlík: Thresholding of a Digital Image by Free Terms", submitted to Journal of Mathematical Imaging and Vision

M.Suganthi and M.Madheswaran: A Novel Approach towards Segmentation of Breast Tumors from Screening Mammograms for Efficient Decision Support System, World Academy of Science, Engineering and Technology 40 2010

Vedoucí diplomové práce: Mgr. Jan Pavlík, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2014/2015.

V Brně, dne 20.11.2014

L.S.

| | |
|---|---|
| _____ | _____ |
| prof. RNDr. Josef Šlapal, CSc. | prof. RNDr. Miroslav Doupovec, CSc., dr. h. c. |
| Ředitel ústavu | Děkan fakulty |

**Abstrakt**
Práce se zabývá metodami segmentace obrazu. Na principu prahování je postavena metoda fuzzy segmentace. Ta je zobecněna do vícekriteriální metody. Celá metoda je podepřena teorií volných algeber. Přes uspořádané množiny na vlastnostech obrazu je vytvořen volný distributivní svaz, jehož prvky jsou termy použitelné k prahování. Je představen rozklad na třídy ekvivalence, které mohou tvořit možné výsledky segmentace. V závěru jsou představeny použité algoritmy a navrženy metody jejich zrychlení. Dále je představena metoda možného odčítání objektů.

**Summary**
The thesis covers methods for image segmentation. Fuzzy segmentation is based on the thresholding method. This is generalized to accept multiple criteria. The whole process is mathematically based on the free algebra theory. Free distributive lattice is created from poset of elements based on image properties and the lattice members are represented by terms used by the threshoding. Possible segmentation results compose the equivalence classes distribution. The thesis also contains description of resulting algorithms and methods for their optimization. Also the method of area subtracting is introduced.

**Klíčová slova**
Digitální prostor, digitální obraz, segmentace obrazu, prahování, fuzzy segmentace, uspořádané množiny, volné algebry, svazy, distributivní svazy, matematické struktury, volné termy, přípustné množiny, algoritmy.

**Keywords**
Digital space, digital image, image segmentation, thresholding, fuzzy segmentation, artially ordered sets, free algebras, lattices, distributive lattices, mathematical structures, free terms, feasible sets, algorithms.

ČAMBALOVÁ, K. *Volné algebraické struktury a jejich využití pro segmentaci digitálního obrazu.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 48 s. Vedoucí Mgr. Jan Pavlík, Ph.D.

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně dle pokynů vedoucího a za použití uvedené literatury.

Bc. Kateřina Čambalová

Děkuji především svému vedoucímu Mgr. Janu Pavlíkovi, Ph.D. za trpělivé a odborné vedení a mnoho cenných rad a podnětů. Dík patří i Mgr. Janu Krihovi za tolik potřebné rady v otázkách programování.

Bc. Kateřina Čambalová

# Obsah

# 1. Úvod

Segmentace je metoda analýzy obrazu, jejímž cílem je najít a vybrat na snímku obrazy skutečných objektů. Vznikl již nespočet metod segmentace založených na různých přístupech, v této práci se však zaměříme především na metody založené na prahování.

Nejlepším segmentačním nástrojem je lidský mozek. Člověk je často schopen v obraze vybrat i objekt, který je velmi špatně odlišen od zbytku, neboť lidský mozek do analýzy obrazu, který oko vidí, dokáže zahrnout i většinu faktických informací, které o dané scéně zná. Mozek automaticky analyzuje tvary, barvy, podobnosti a především vyniká velmi vysokou schopností dedukce. Jakákoliv nápodoba lidského mozku je v současné době počítačově prakticky nemožná, neboť se jedná o strukturu natolik komplexní, že je technicky nesestavitelná. Jediné, o co se můžeme pokoušet, je vytvoření pouhé zjednodušené aproximace tak složitého systému.

Přestože v minulosti bylo použitelnou metodou segmentace v některých kritických aplikacích dokonce i posazení vědce s fixou mezi tiskárnu a skener, v současné době je naší snahou tento proces co nejvíce zautomatizovat.

Ve druhé kapitole se seznámíme s pojmy a konstrukcemi digitálního prostoru. Objasníme si jeho souvislost s digitálním obrazem a uvedeme, proč je tak vhodný pro popis digitálních snímků. Dále si vysvětlíme, co je cílem segmentace a představíme zcela fundamentální metodu, prahování. Na jejím základě dále postupně vystavíme teorii fuzzy segmentace, kde se poprvé ukáží některá tvrzení podávající základ dalšímu vývoji. Zde je představen pojem fuzzy spjatosti a poprvé použijeme vzájemné podobnosti v obraze k tomu, abychom definovali segmentovaný objekt.

Pro další rozvoj však bude nezbytné položit některé teoretické základy, abychom sestavili korektní aparát k vývoji dalších metod. Tímto se zabývá třetí kapitola, která poskytuje hlubší náhled do některých částí algebry, předkládá propojení konkrétních pojmů ve snaze ukázat všechny souvislosti, které vedly k sestavení představené metody, a podává náhled na další směřování teoretických úvah. Zde budujeme nutný podklad, aby bylo možno provádět segmentaci založenou na podobnosti částí obrazu, která již ale nebude omezena na analýzu jediné jeho vlastnosti.

Algebraická část vychází z pojmů uspořádaných množin, plynule přejde k představení svazů a jejich konkrétních vlastností, které se ukáží být velmi užitečnými pro požadované zobecnění. Zcela zásadním se pro naše účely ukáže distributivní svaz. S pomocí některých speciálních prvků a konstrukcí ve svazech vystavíme prostor, kde bude probíhat řešení segmentace. Zároveň nastíníme jiný pohled na uvedenou teorii s pomocí matematických struktur, které nám mohou pomoct k lepšímu pochopení použitých konstrukcí.

Čtvrtá kapitola je věnována samotné metodě rozšířené, vícekriteriální, segmentace. Vyjdeme z analýzy prvků dříve představeného prostoru. Ukáže se, že ten je vytvořen nad množinou jistých vlastností původního obrazu, pomocí jehož prvků pak definujeme vzájemnou podobnost či odlišnost částí obrazu. Přineseme popis segmentovatelného objektu pomocí přípustných množin, které zároveň mohou pomoci k jeho lepší charakteristice. V závěru kapitoly odvodíme a představíme algoritmus hledání vhodného prahu, který tvoří

jádro jakékoliv praktické aplikace. Prodiskutujeme zde i otázku jeho složitosti.

V poslední kapitole jsou uvedeny některé poznámky k implementaci uvedené teorie. Jsou zde představeny konkrétní algoritmy jak pro jednoduchou fuzzy segmentaci, tak pro navazující vícekriteriální metodu. Základními algoritmy navazující metody je algoritmus výstavby fuzzy oblastí a navazující algoritmus získávání hodnot pro stavbu prahu. Je zde nastíněn problém jejich složitosti a jsou navržena některá vylepšení. Předně u druhého z algoritmů se jedná o dva možné přístupy jejich zrychlení, porovnáme zde také jejich efektivnost.

Nakonec je představeno možné rozšíření současné vícekriteriální metody. To je založeno na vložení více empirických znalostí do problému. Pokusíme se představit možnosti vyloučení některých bodů z výsledného objektu.

# 2. Segmentace digitálního obrazu

Segmentace je základní metodou zpracování digitálního obrazu, jejímž cílem je rozdělit obraz na jednotlivé části, které odpovídají reálným objektům na obraze, proto takovým částem též říkáme *objekty*. Případně můžeme chtít vybrat jediný takový objekt, jež je předmětem našeho zájmu. Největšího užití se segmentaci obrazu dostává při zpracování různých medicínských snímků, například z počítačové tomografie, či v konstrukci expertních systémů a inteligentních systémů, u nichž je snaha vytvořit jistý typ vnímání okolí, tzv. počítačové vidění. Rozvoj metod segmentace jde ruku v ruce s vývojem metod zpracování obrazu a i samotných digitálních technologií. Existuje skutečně nespočet metod segmentace, od velmi jednoduchých, pracujících pouze s lokálními vlastnostmi obrazu, přes metody statistické, uvažující i vlastnosti obrazu v okolí posuzovaného bodu, až po metody velmi složité pracující například na principech frekvenční analýzy.

## 2.1. Digitální prostor

Zpracování digitálního obrazu začíná již jeho pořízením. Jelikož většina současných digitálních záznamových zařízení snímá obraz pomocí světlocitlivých čipů s jednotlivými segmenty uspořádanými do pravidelné mřížky, výsledný obraz tak nemůže být spojitý. Každému bodu mřížky pak přísluší nějaká aproximace hodnoty barevného světla, které dopadlo na plochu příslušného segmentu čipu.

Definujme *digitální prostor* jako dvojici $(V, \pi)$, kde $V$ je neprázdná množina bodů a $\pi$ je symetrická binární relace *přilehlosti*, taková, že množina $V$ je souvislá vzhledem k $\pi$. Definice je to velmi obecná, v praxi ale budeme nejčastěji používat především čtvercovou mřížku, která je důsledkem povahy snímacího čipu. Pravidelná $N$-rozměrná mřížka je množina bodů

$$(\delta\mathbb{Z})^N = \{(\delta c_1, \cdots, \delta c_N) \mid c_n \in \mathbb{Z},\ 1 \leq n \leq N\}.$$

*Čtvercová mřížka* je konkrétním případem takové mřížky pro dvourozměrný prostor a jednotkovou vzdálenost $\delta$, tedy $\mathbb{Z}^2$. Je to mřížka pro aplikace nejvýznamnější, neboť tuto strukturu má většina digitálních snímků. Mřížka je vždy diskretizací reálného prostoru, takže každému jejímu bodu můžeme přiřadit reálné okolí, takzvanou Voronného buňku, což je množina všech bodů, které jsou k danému bodu mřížky blíže, než k ostatním bodům mřížky. Voroného buňkám ve čtvercové mřížce říkáme *pixely* (zkratka termínu picture element). Další prakticky významnou mřížkou je $\delta\mathbb{Z}^3$. Tato takzvaná *kubická mřížka* najde uplatnění právě například v počítačové tomografii, kde jsou snímány dvourozměrné řezy se vzájemnými rozestupy $\delta$. Jejich složením pak dostaneme trojrozměrný obraz. Voronného okolím v tomto případě říkáme *voxely*.

Tyto elementy spolu sousedí, což teoreticky popíšeme právě relací přilehlosti. Obecně by mohla být přilehlostí jakákoliv relace $\pi \subset V^2$, my na ni ale máme další požadavky. Požadavek symetrie je poměrně přirozený, protože pokud element $x$ sousedí s $y$, je vcelku rozumné požadovat, aby také $y$ sousedil s $x$. Požadavek, abychom pracovali na prostoru jistým způsobem souvislém, je též velmi přirozený.

**Definice 2.1.** *Posloupnost $\langle x_0, x_1, \cdots, x_n\rangle$ prvků z $V$ takovou, že $x_0 = x$, $x_n = y$ a $(x_{k-1}, x_k) \in \pi$, $1 \leq k \leq n$, nazveme* $\pi$-cesta *ve $V$ spojující $x$ a $y$, kde $n$ nazveme* délkou cesty.

Souvislost $V$ vzhledem k $\pi$ znamená, že lze každé dva prvky $x, y \in V$ spojit $\pi$ cestou. Relace spojitosti $x$ a $y$ $\pi$-cestou je zřejmě reflexivní, stačí vytvořit triviální cestu $\langle x \rangle$ délky 0, a také tranzitivní. Pokud totiž existuje cesta $\langle x = x_0, \cdots, x_n = y \rangle$ mezi $x$ a $y$ a cesta $\langle y = y_0, \cdots, y_n = z \rangle$ mezi $y$ a $z$, pak existuje alespoň cesta $\langle x = x_0, \cdots, x_n = y_0, \cdots, y_n = z \rangle$ z $x$ do $z$. Pokud je navíc přilehlost $\pi$ symetrická, je symetrická relace spojitosti $\pi$-cestou.

V pravidelné mřížce $\mathbb{Z}^N$ jsou dvě nejvýznamnější relace přilehlosti $\omega_N$ a $\alpha_N$, definované jako

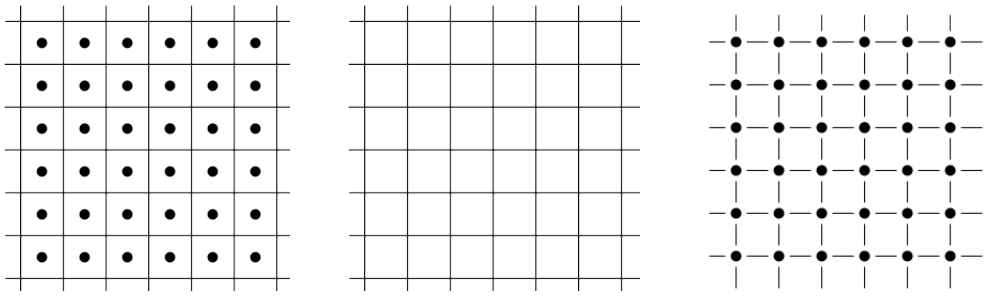
$$(x, y) \in \omega_N \Leftrightarrow \sum_{n=1}^{N} |x_n - y_n| = 1$$

$$(x, y) \in \alpha_N \Leftrightarrow x \neq y \text{ a } |x_n - y_n| \leq 1,\ 1 \leq n \leq N$$

Ve dvourozměrném prostoru říkáme přilehlosti $\omega_2$ stranová přilehlost nebo také 4přilehlost, protože za přilehlé považujeme pixely, jež sdílí stranu. Každý pixel tak má 4 pixely přilehlé. Přilehlosti $\alpha_2$ říkáme 8přilehlost. Pixelům stačí sdílet vrchol nebo stranu, proto je kolem každého pixelu osm k němu přilehlých. Ve třech dimenzích je $\omega_3$ opět stranovou přilehlostí a $\alpha_3$ je přilehlostí stranově-hranovou.

Jak mřížek, tak přilehlostí existuje mnohem více. Různé typy, stejně jako jejich různé vlastnosti, vhodnost a možné isomorfismy hlouběji rozebírá [5, především kap. 1 a 2].

Z předchozího je zřejmé, že digitální prostor $(V, \pi)$ tvoří spojitý neorientovaný graf, kde $V$ jsou uzly a $\pi$ je množina hran. Po hranách se lze přesouvat z jednoho uzlu do druhého a vytvořit tak cestu. V některých chvílích na něj tedy budeme nahlížet spíše v grafovém pojetí a budeme tak moci použít některé z metod teorie grafů.



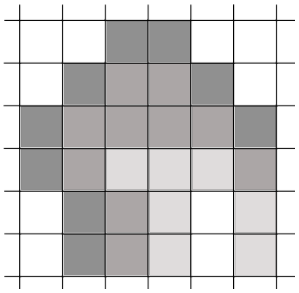
Obrázek 2.1: *Vlevo je znázorněn digitální prostor v rovině jako body mřížky, vlevo a uprostřed jsou navíc zvýrazněny buňky příslušející každému bodu, takzvané pixely. Vpravo je digitální prostor znázorněn jako graf. Body prostoru tvoří uzly, spojnice jednotlivých uzlů tvoří hrany grafu.*

### 2.1.1. Obraz v digitálním prostoru

Digitální prostor nám pomohl formalizovat povahu záznamového zařízení, ale protože při pořízení obrazu jsme zaznamenali nějaké hodnoty barevného světla, budeme je v tomto prostoru také chtít nějak zapsat. Vytvoříme si proto digitální obraz.

**Definice 2.2.** Digitální obraz *nad digitálním prostorem* $(V, \pi)$ *je trojice* $(V, \pi, f)$, *kde* $f : V \to \mathcal{C}$ *je funkce přiřazující bodům z* $V$ *hodnoty z množiny barev* $\mathcal{C}$.

Množina barev $\mathcal{C}$ by mohla značit například stupně šedi, nebo při $\mathcal{C} = \mathbb{R}^3$ by určovala hodnoty pro každý ze tří barevných kanálů, červený, zelený a modrý. Takto už můžeme popsat většinu obvyklých obrazů, jaké známe třeba z monitoru počítače.



Obrázek 2.2: *Digitální prostor, kde jednotlivé pixely mají přiřazeny barevné hodnoty.*

Je-li barva v obraze nějaký prvek $\mathbb{R}^n$, lze ji zapsat jako uspořádanou $n$-tici. Některé barvy lze porovnávat. Například lze říci, že černá barva ve standardu RGB, kterou zapíšeme trojicí $(0, 0, 0)$, je tmavší než světle žlutá, jež je reprezentována jako $(255, 255, 150)$. Nyní si takové uspořádání blíže specifikujeme.

**Definice 2.3.** *Mějme uspořádané n-tice* $(a_1, a_2, \cdots, a_n)$ *a* $(b_1, b_2, \cdots, b_n)$*. Jejich uspořádání zavedeme jako*

$$(a_1, a_2, \cdots, a_n) \leq (b_1, b_2, \cdots, b_n) \;\Leftrightarrow\; a_1 \leq b_1 \;\wedge\; a_2 \leq b_2 \;\wedge \cdots \wedge\; a_n \leq b_n,$$

$$(a_1, a_2, \cdots, a_n) = (b_1, b_2, \cdots, b_n) \;\Leftrightarrow\; a_1 = b_1 \;\wedge\; a_2 = b_2 \;\wedge \cdots \wedge\; a_n = b_n.$$

Z tototo můžeme odvodit i ostré uspořádání $(a_1, a_2, \cdots, a_n) < (b_1, b_2, \cdots, b_n) \;\Leftrightarrow\; (a_1, a_2, \cdots, a_n) \leq (b_1, b_2, \cdots, b_n) \;\wedge\; (a_1, a_2, \cdots, a_n) \neq (b_1, b_2, \cdots, b_n)$.

Na tento obraz ale vzneseme ještě jeden praktický požadavek. Zpravidla budeme pracovat nad obrazy konečnými, to znamená, že nosič funkce $f$ je konečný. Jinými slovy množina $\{x \in V \mid f(x) > \mathbf{o}\}$ má konečný počet prvků. Prvkem $\mathbf{o}$ zde myslíme jistý nulový prvek množiny $\mathcal{C}$, lze jej například považovat za černou barvu pozadí, které však při segmentaci nebude přímo předmětem našeho zájmu.

Nadále budeme uvažovat především dvourozměrné obrazy s pravidelnou mřížkou, avšak některá teorie je přenositelná i na prostorové obrazy.

**Definice 2.4.** Binární obraz *nad digitálním prostorem* $(V, \pi)$ *je trojice* $(V, \pi, f)$*, kde* $f : V \to \langle 0,\ 1\rangle$.

Zde je množina barev pouze dvouprvková, proto obraz nazýváme binární. Lze si jej představit jako obraz černobílý, kde nulové pixely tvoří černé pozadí a pixely s hodnotou 1 tvoří bílý objekt v obraze.

## 2.2. Jednoduché metody segmentace

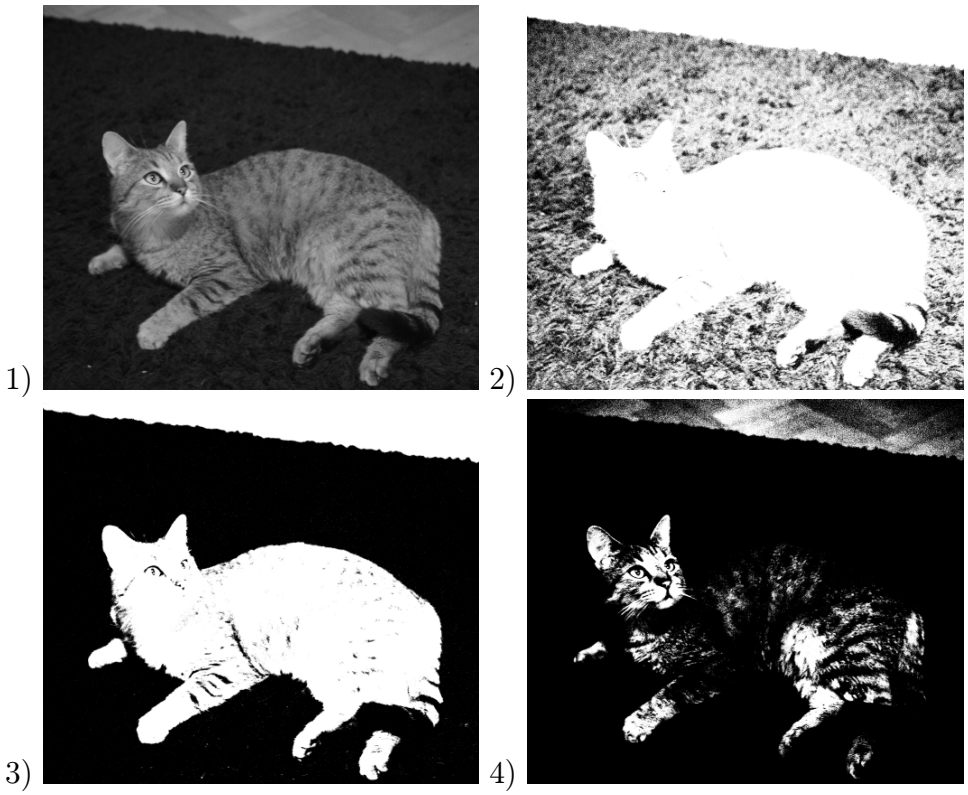
Segmentací se snažíme rozdělit obraz na objekty, tím zde rozumíme oblasti obrazu, které odpovídají reálným objektům zachyceným na obraze. Taková oblast segmentace by neměla být podmnožinou žádného většího objektu a ani by se s jinou oblastí neměla protínat, kromě nejvýše jejich společné hranice.

### 2.2.1. Prahování

Výsledkem segmentace, kterého se zde pokusíme dosáhnout, je převedení barevného obrazu na binární. Chceme tedy rozdělit obraz na dvě části, kde pixely s hodnotou 1 budou značit hledaný objekt, zatímco nulové pixely budeme považovat za pozadí. Nejjednodušší z metod nalezení takového obrazu je bezesporu prahování. Jedná se o metodu lokální,která posuzuje každý pixel zvlášť a o tom, zda patří objektu, rozhodne jen na základě jeho barvy a uživatelem zadaného kritéria. Toto kritérium označíme $t$ a nazveme jej *práh*. Výsledkem bude obraz $(V, \pi, f_t)$, kde funkce $f_t$ přiřazuje pixelům hodnoty 0 nebo 1 následovně

$$f_t(x) = \begin{cases} 1, & \text{pro } f(x) \geq t \\ 0, & \text{pro } f(x) < t. \end{cases}$$

Je zřejmé, že tuto metodu lze aplikovat jen na jednobarevný obraz, tedy jen ten, kde $\mathcal{C} = \mathbb{R}$. Pokud bychom chtěli prahovat barevný obraz, museli bychom z něj vybrat pouze jedinou vlastnost, například úrovně jasu nebo jen jeden z barevných kanálů, nebo jej nejprve převést na obraz ve stupních šedi.




Obrázek 2.3: *Ukázka prahování na obraze, kde je objekt na kontrastním pozadí. 1) původní obraz, 2) příliš nízký práh, 3) optimální práh, 4) příliš vysoký práh.*

Prahování funguje dobře jen pro obrazy, kde je objekt umístěn na dostatečně kontrastním pozadí, například světlý objekt na tmavém pozadí, nebo obráceně. V těchto velmi konkrétních případech je prahování postačující, poměrně efektivní a lze snadno automatizovat. Histogram takového obrazu je obvykle tvořen dvěma vrcholy oddělenými významným minimem. Optimálním prahem je hodnota, kde leží toto minimum. Vhodně kontrastní obraz je na obrázku 2.3. Zde vidíme závislost prahování na výběru vhodného

prahu. Na 2) je použit práh příliš nízký, kromě hledaného objektu obsahuje prahovaný objekt příliš velkou oblast pozadí objektu. V 3) byl zvolen vhodný práh dle histogramu, objekt je zde dobře vymezen vůči tmavému pozadí, ale vidíme, že pokud obraz obsahuje jiné světlé oblasti, pak jsou také zahrnuty do prahovaného obrazu, což je dáno tím, že metoda nebere v úvahu žádnou souvislost objektu. Pro 4) jsme naopak použili práh příliš vysoký, proto hledaný objekt není zachycen celý, přesto je stále do výsledného objektu zahrnuta část světlejší oblasti v pozadí.

1)  2) 

Obrázek 2.4: *1) Obraz nevhodný pro prahování, 2) zde výsledek prahování neodpovídá žádnému reálnému objektu, [Dwight Hooker].*

Pro obecné obrazy ale může tato metoda být i velmi neefektivní, nedokáže si poradit například se stínováním, ani odlišit více objektů podobné barvy. Jak je vidět na obrázku 2.2, ani pokud umístíme práh do nějakého lokálního minima histogramu, tak není zaručeno, že nějaký objekt vybereme celý.

### 2.2.2. Segmentace založená na afinitě

Hlavní nevýhody prahování pramení z toho, že metoda nebere v úvahu, že reálný objekt bývá zpravidla spojitý. Je rozumné očekávat, že dva body objektu, které k sobě leží relativně blízko, by si měly být i podobné. Takto lze zajistit, aby například pomalu stínovaný objekt nebyl segmentací rozdělen na dva.

**Definice 2.5.** Afinita *v digitálním prostoru* $(V, \pi)$ *je funkce* $\psi : V^2 \to \langle 0, 1\rangle$ *splňující*

1. $\forall x \in V,\ \psi(x, x) = 1$
2. $\forall x, y \in V,\ \psi(x, y) = \psi(y, x)$

*[2, str. 74]*

Afinita popisuje, jak jsou si dva pixely podobné, jak moc jsou mezi sebou spjaté. Dá s očekávat, že pixely ležící ve stejném objektu budou spjaté ve velké míře. Pomocí afinity $\psi(x, y)$ pak lze popsat pravdivost výroku

$$x \in X \Leftrightarrow y \in X.$$

Vlastně se jedná o míru jistoty, že pixely jsou ve stejném objektu $X$. Zde se objevuje první podobnost s teorií fuzzy množin. Fuzzy množina $\tilde{A}$ je definována jako dvojice $\tilde{A} = (X, \mu)$, kde $X$ je nosná množina bodů a $\mu : X \to \langle 0, 1\rangle$ je míra, jež vyjadřuje stupeň příslušnosti každého bodu k fuzzy množině. Stupeň příslušnosti 1 znamená naprostou jistotu, že bod k dané množině patří, naopak 0 značí jistotu, že pixel k fuzzy množině nepatří. Podobně zde 1 vyjadřuje, že oba pixely nepochybně leží ve stejném objektu. Proto je přirozený požadavek, aby afinita jednoho a téhož pixelu byla nejvyšší možná. Díky této podobnosti obou teorií používáme pro afinitu termín *fuzzy afinita*.

Zatím je funkce $\psi$ definována docela obecně, konkrétní podobu funkce totiž vybíráme až podle aplikace. Afinita pak může být funkce závislá na rozdílech barev pixelů, jejich vzdálenosti, na nějakých statistických vlastnostech z okolí obou pixelů nebo i na několika takových vlastnostech zároveň. Často budeme na afinitu klást ještě jeden požadavek, a to nulovost pro nepřilehlé pixely

$$\psi\left(x, y\right) = 0 \quad \text{pro} \quad (x, y) \notin \pi.$$

Žádná z uvedených vlastností ale nezaručuje tranzitivitu afinity. Představme si nějakou $\pi$-cestu, kde každé dva po sobě jdoucí pixely mají vysokou afinitu, například je jeden pixel jen o málo světlejší, než následující, ale první a poslední pixel si mohou být podobné velmi málo, první může být velmi světlý, zatímco poslední bude tmavý. Tak by se mohlo stát, že bychom první a poslední pixel ke stejnému objektu nepřiřadili. Naštěstí zaručení takové tranzitivity není v naší situaci třeba. Pro účely segmentace bude postačující funkce, jež zajišťuje alespoň tranzitivitu příslušnosti ke stejnému objektu. Pokud jsou pixely $x$ a $y$ s vysokou mírou afinity v jednom objektu a pokud jsme si také s vysokou mírou jisti, že $y$ a $z$ jsou ve stejném objektu, chtěli bychom, aby i $x$ a $z$ náležely s vysokou mírou pravdivosti jednom objektu.

Dva pixely mohou být spojeny různými $\pi$-cestami, těm můžeme říkat *řetězce*, a jednotlivým dvojicím přilehlých pixelů $(x_{k-1}, x_k)$ budeme říkat *články* řetězce. Na základě afinity vytvoříme novou funkci, která bude brát v úvahu různé možnosti jak se dostat z pixelu $x$ do pixelu $y$.

**Definice 2.6.** Fuzzy spjatost *je funkce* $\mu : V^2 \to \langle 0, 1\rangle$ *definovaná předpisem*

$$\mu(x, y) = \max_{\langle x_0 = x, x_1, \cdots, x_n = y\rangle} \min_{0 \le k \le N} \psi\left(x_{k-1}, x_k\right)$$

*[2, str. 74]*

Afinitu $\psi\left(x_{k-1}, x_k\right)$ si lze představit jako sílu článku. Řetězec je pak jenom tak silný, jako jeho nejslabší článek, což při intuitivním pohledu není příliš překvapivé. Takže sílu řetězce realizujeme minimem přes všechny jeho články. Pak vybereme nejsilnější z řetězců mezi $x$ a $y$, to realizujeme maximem přes všechny možné cesty.

Nyní uvažujme relaci existence takové cesty mezi dvěma body. Relace je zřejmě reflexivní a symetrická. Dále pokud $\mu\left(x, y\right) \ge t$ a $\mu\left(y, z\right) \ge t$, pak existuje alespoň jeden řetězec z $x$ do $z$, jehož stupeň fuzzy spjatosti je alespoň $t$, vždy totiž můžeme spojit oba zmíněné řetězce v bodě $y$ a pořád nebude síla žádného článku menší než $t$. Takže platí tranzitivita příslušnosti ke stejnému objektu.

**Věta 2.1.** *Pro daný práh* $t \in \langle 0, 1\rangle$ *a danou fuzzy spjatost* $\mu$ *je binární relace* $K_{\mu,t}$ *definovaná jako*

$$(x, y) \in K_{\mu,t} \Leftrightarrow \mu\left(x, y\right) \ge t$$

*relací ekvivalence.* *[2, str. 74]*

Se zadaným prahem a fuzzy spjatostí se obraz rozpadne do tříd ekvivalence. Tyto třídy pro nás budou poměrně významné, neboť celá metoda spočívá v hledání nějaké takové vhodné třídy. Proto je také pojmenujeme jako *fuzzy souvislé objekty* nebo jednoduše *fuzzy objekty*. Jejich nejvýznamnější vlastností je jejich spjatost. Pokud je afinita nulová pro nepřilehlé pixely, tedy $\psi(x, y) = 0$ pro $(x, y) \notin \pi$, pak jsou fuzzy objekty spojité vzhledem k přilehlosti $\pi$. Množství vzniklých fuzzy objektů závisí na zadaném prahu. Bude-li práh malý, vznikne menší počet tříd ekvivalence, které budou velké, bude-li naopak velký, vznikne větší počet malých objektů.

Protože nás však často nezajímá rozdělení obrazu na všechny fuzzy objekty, ale chceme spíše vybrat jeden hlavní objekt zájmu, raději najdeme nějaký pixel $u$, o němž jsme přesvědčeni (například na základě našeho odborného odhadu), že k tomuto objektu patří. Pak se pokusíme najít všechny pixely, které nejspíš patří stejnému objektu. Nyní vytvoříme funkci $m_u : x \mapsto \mu(u, x)$, jež každému bodu přiřadí míru fuzzy spjatosti s námi vybraným pixelem $u$. Nad původním digitálním prostorem takto vytvoříme nový obraz $(V, \pi, m_u)$, který nazveme *spjatostní mapa*. Na této mapě nyní můžeme provést prahování. Tím se obraz rozdělí na dvě skupiny, na fuzzy objekt $\{x \in V \mid \mu(x, u) \geq t\}$ a pozadí.

*Třídu ekvivalence* značíme $[x]_\sim = \{y \in X \mid x \sim y\}$ a tvoří ji všechny prvky nosné množiny$X$, které jsou ekvivalentní se zadaným prvkem $x$. Zde třídu ekvivalence obsahující prvek $u$ tvoří právě všechny pixely $x$, jež jsou si s $u$ dostatečně podobné, tedy $[u]_{K_{\mu,t}} = \{x \in X \mid \mu(u, x) \geq t\}$, což je zřejmě tentýž fuzzy objekt, který jsme vytvořili prahováním. Pokud bychom navíc vytvořili spjatostní mapu pro kterýkoliv jiný prvek tohoto objektu, dostali bychom prahováním stejný fuzzy objekt, neboť jsou si všechny prvky v objektu v tomto ohledu ekvivalentní.

V zájmu zdůraznění souvislostí zde ještě uvedeme, že v teorii fuzzy množin prahování odpovídá *$\alpha$-řez*, což je ostrá (nefuzzy) množina $A_\alpha = \{x \in X \mid \mu(x) \geq \alpha\}$.

Některé fuzzy objekty budeme považovat i v další teorii za natolik významné, že si pro ně zavedeme zvláštní označení.

**Definice 2.7.** $\Theta(x, t) = \{x \in X \mid \mu(u, x) \geq t\}$ *označíme množinu všech $y$ spjatých s $x$ alespoň s mírou $t$.* *[6, str. 340]*

Pokud ovšem hledáme v obraze nějaký objekt, může být velmi složité předem odhadnout velikost prahu. Nebylo by tedy jednodušší vybrat více bodů a vytvořit takový objekt, který všechny tyto body obsahuje? Jednodušší, než zkoušet různé hodnoty prahu a pozorovat, zda „strefíme“ hledaný objekt, bude například vybrat nějaký velmi typický pixel a další pixel, jež je mu co nejméně podobný, ale u kterého jsme si stále velmi dobře jisti jeho příslušností k objektu. Vybereme-li tedy dva body a požadujeme, aby oba náležely jednomu objektu, z předchozího víme, že budou vzhledem k příslušnosti k tomuto objektu ekvivalentní. Vybereme-li tedy jakýkoliv dostatečně nízký práh a vytvoříme pro oba body příslušné fuzzy objekty, budou oba objekty stejné. Jak ale máme jednoznačně určit, jaký práh vybrat? Pokud budeme chtít, aby výsledný objekt obsahoval co nejméně bodů, které k němu ve skutečnosti nepatří, je rozumné, abychom hledali objekt co nejmenší. Proto jako práh zvolíme právě míru spjatosti obou vybraných bodů. Jakýkoliv větší práh by způsobil, že každý z těchto dvou pixelů by ležel v jiné třídě rozkladu a výsledný objekt by obsahoval jen jeden z nich.

**Definice 2.8.** *Nejmenší fuzzy spjatý objekt obsahující body $x$ a $y$ označíme jako* $\Omega(x,y) = \Theta(x,\mu(x,y)) = \Theta(y,\mu(y,x))$. *[6, str. 340]*

Všechny pixely, které se do objektu $\Omega$ dostanou, musí být s každým dalším pixelem spjaty s mírou alespoň $\mu(x,y)$, což dává vzniknout ternární *relaci spjatosti* $\Phi$

$$\Phi(x,y,z) \Leftrightarrow z \in \Omega(x,y).$$

Tuto relaci lze přepsat do pojmů fuzzy spjatosti následovně

$$\Phi(x,y,z) \;\Leftrightarrow\; \mu(x,z) \geq \mu(x,y) \;\wedge\; \mu(y,z) \geq \mu(x,y).$$

Zkoumáním relace $\Phi$ odvodíme základ celé následující metody.

Aplikace této metody vyžaduje algoritmus, jež bude prohledávat celý obraz, dokud nebude zaručeno, že jsme pro všechny pixely našli minimální hodnotu fuzzy spjatosti. Algoritmus, který toto řeší, dostaneme úpravou Dijkstrova algoritmu pro hledání minimální cesty v grafu. Použití tohoto algoritmu je možné díky grafové struktuře digitálního prostoru. Přilehlé pixely považujeme za uzly spojené hranou, jejímž ohodnocením je hodnota afinity těchto dvou pixelů.

Dijkstrův algoritmus začíná z výchozího uzlu, pro každý navštívený uzel si pamatuje zatím nejnižší dosaženou hodnotu cesty a přechází do nenavštívených uzlů vždy po hraně s nejnižší hodnotou. Jeho složitost se pohybuje v řádu $\mathcal{O}(|V^2|)$.

Modifikace pro hledání spjatostní mapy pracuje spíše se seznamem navštívených pixelů, než se seznamem těch nenavštívených, a hledá maximální cestu. Pokračuje vždy z pixelu s nejvyšší hodnotou $m_u$ a porovná, zda se zlepší hodnota cesty do všech přilehlých pixelů.

V přiloženém programu je provedena jednoduchá implementace hledání spjatostní mapy s afinitou

$$\psi(x,y) = \frac{2}{1 + |f(x) - f(y)|} - 1,$$

kde $f(x)$ je hodnota jasu v pixelu $x$ normovaná na hodnotu z rozsahu $\langle 0, 1\rangle$.

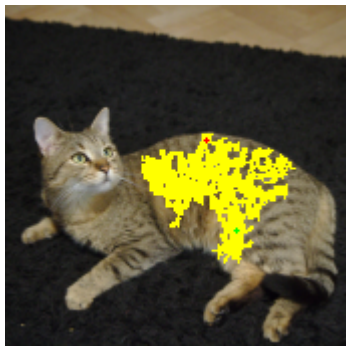
Na obrazech rozumných rozměrů je metoda poměrně rychlá a v některých případech dává velmi dobré výsledky. Je ale značně náchylná na vhodný výběr výchozích pixelů. Je-li objekt jen málo oddělen od pozadí nebo se na jeho ploše příliš rychle mění hodnota jasu, hrozí, že segmentovaný objekt „přeteče“ do pozadí.

Hlavními nevýhodami je však možnost porovnávání pouze jedné vlastnosti, například jasu či jediného barevného kanálu. Navíc je naprosto nemožné takto vyhledat celý objekt, pokud je z jakéhokoliv důvodu nesouvislý.

Na následujících obrázcích je znázorněn výsledek segmentace. Výchozí bod je označen červeně, bod, podle kterého nastavujeme hodnotu prahu, značíme zeleně. Uvádíme zde i příslušnou spjatostní mapu $(V, \pi, m_u)$. Bílá barva pixelu znamená, že je s výchozím spjat s fuzzy spjatostí $\mu(x,u) = 1$, tmavší barva značí nižší hodnotu spjatosti.

Na obrázku 2.5 je vidět, že vzhledem k poměrně necitlivému kritériu jsou v mapě viditelné poměrně malé rozdíly i velmi vzdálených pixelů. Na spodním segmentovaném obraze je vidět, že o málo odlišná volba vstupních pixelů způsobila výběr pozadí.

Naopak na obrázku 2.6 je patrné značné zlepšení oproti předchozí metodě, také jsou o něco lépe zřetelné odlišnosti ve spjatostní mapě.

1)  2)  3) 

4) 

Obrázek 2.5: *1) Původní obraz, 2) spjatostní mapa a 3) výsledek segmentace, 4) výsledek segmentace pro jiné vstupní body.*

1)  2) 

Obrázek 2.6: *1) Spojitostní mapa a 2) výsledek segmentace.*

# 3. Algebraické pozadí vícekriteriální segmentace

## 3.1. Uspořádané množiny

Afinní metoda pracovala s intervalem $\langle 0, 1\rangle$. K porovnání fuzzy spojitosti s prahem jsme využili vlastnosti uspořádání, jehož znalost je však na reálných číslech téměř triviální. Nadále ale budeme porovnávat prvky složitější struktury, proto zavedeme uspořádání pečlivěji.

**Definice 3.1.** *Buď $M$ množina a $\leq$ binární relace na $M$ splňující*

- *reflexivitu:* $\forall x \in M \ x \leq x$
- *antisymetrii:* $\forall x, y \in M \ (x \leq y \wedge y \leq x) \Rightarrow x = y$
- *tranzitivitu:* $\forall x, y, z \in M \ (x \leq y \wedge y \leq z) \Rightarrow x \leq z$,

*pak $\leq$ nazýváme* uspořádání *(nebo také* částečné uspořádání*) na $M$ a dvojici $(M, \leq)$ nazveme* (částečně) uspořádaná množina*.*

*Platí-li navíc*

- *srovnatelnost:* $\forall x, y \in M \ x \leq y \vee y \leq x$,

*pak se $(M, \leq)$ nazývá* lineárně uspořádaná množina*.* *[7, str. 29]*

Vidíme, že v předchozí metodě tvořily hodnoty fuzzy spojitosti lineárně uspořádanou množinu, proto mělo smysl se pro každou dvojici pixelů ptát, zda je fuzzy spojitost větší či menší než zadaný práh, neboť díky srovnatelnosti je takové porovnání pokaždé možné. I proto lze předchozí metodu označovat jako lineární. Naopak v částečně uspořádané množině mohou existovat *nesrovnatelné prvky*. Zkrátka nelze říci, který ze dvou takových prvků je větší a který menší. Ale i pokud jsou některé prvky navzájem nesrovnatelné, vždy mohou existovat prvky jiné, se kterými je porovnat lze.

**Definice 3.2.** *Buď $(M, \leq)$ částečně uspořádaná množina. Pak se $m \in M$ nazývá* maximální *(resp.* minimální*) prvek množiny $M$ právě tehdy, když $\forall x \in M$ platí $x \geq m$ (resp. $x \leq m$ ) $\Rightarrow x = m$ .* *[7, str. 29]*

Každý prvek, alespoň v našem případě, kdy se budeme kvůli konečnému obrazu omezovat na konečné množiny, je porovnatelný s některým maximálním. Ač se jedná o užitečnou vlastnost, často je vhodnější, pokud lze všechny prvky porovnávat jen s jediným takovým. Maximální prvky však jednoznačně určené nejsou.

**Definice 3.3.** *Buď $(M, \leq)$ částečně uspořádaná množina. Pak se $\top \in M$ nazývá* největší *(resp. $\bot \in M$ se nazývá* nejmenší*) prvek množiny $M$ právě tehdy, když $\forall x \in M$ platí $\top \geq x$ (resp. $\bot \leq x$).* *[7, str. 29]*

Pokud už tedy takový největší prvek existuje, je zřejmě jediný a navíc bude zároveň jediným maximálním (resp. minimálním). Tím pádem každý jiný prvek musí být menší, než tento největší. Obdobně pro nejmenší prvek. V intervalu $\langle 0, 1\rangle$ z předchozí metody je nejmenším prvkem nepřekvapivě 0 a největším 1.

**Definice 3.4.** *Nechť je $(M, \leq)$ částečně uspořádaná množina a $N \subseteq M$. Pak prvek $u \in M$ nazveme horní závorou množiny $M$ právě tehdy, když $\forall x \in N : \ u \geq x$. Nejmenší prvek množiny všech horních závor množiny $N$ nazveme supremum množiny $N$ a označíme jej* $\sup N$ *nebo* $\bigvee N$.
*Prvek $v \in M$ nazveme dolní závorou množiny $M$ právě tehdy, když $\forall x \in N : \ v \leq x$. Největší prvek množiny všech dolních závor množiny $N$ nazveme infimum množiny $N$ a označíme jej* $\inf N$ *nebo* $\bigwedge N$. *[7, str. 30]*

Supremum ani infimum podmnožiny v ní nemusí ležet, ale pokud existuje, tak i přestože budou prvky podmnožiny navzájem nesrovnatelné, získáme nějaký prvek, se kterým je lze všechny porovnat. Tvrzení pro infima jsou obdobná, neboť infima a suprema jsou pojmy navzájem duální. *Dualita* pojmů způsobí, že pokud převrátíme uspořádání a vzájemně zaměníme všechna infima a suprema v platném tvrzení, toto tvrzení zůstane platným.

S uspořádáním souvisí i dva významné typy množin

**Definice 3.5.** *Nechť $P$ je uspořádaná množina a $Q \subseteq P$.*

- *$Q$ je* horní množina*, pokud pro každé $x \in Q$ a $y \in P$ takové, že $x \leq y$ platí $y \in Q$*
- dolní množina *je duální pojem k horní množině.*

*[3, str. 20]*

Do horní množiny spadají s nějakým prvkem i všechny větší, do dolní množiny zase všechny menší. Zkrátka pokud se v dolní množině budeme pohybovat z nějakého prvku kamkoliv směrem k menším, dolní množinu nemůžeme opustit. Horní a dolní podmnožiny generované neprázdnou podmnožinou $Q$ uspořádané množiny $P$:

$$\uparrow Q = \{y \in P \mid \exists x \in Q, \text{ že } y \geq x\} \quad \text{a} \quad \downarrow Q = \{y \in P \mid \exists x \in Q, \text{ že } y \leq x\}.$$

Horní a dolní podmnožiny generované prvkem $x \in P$:

$$\uparrow x = \{y \in P \mid y \geq x\} \quad \text{a} \quad \downarrow x = \{y \in P \mid y \leq x\}.$$

## 3.2. Zobecnění lineární metody

Předchozí metoda vycházela z faktu, že všechny pixely lze lineárně uspořádat podle podobnosti s výchozím bodem. Pak už je snadné podle prahu rozdělit možné podobnosti a tím pádem i všechny pixely na dvě skupiny. To je ale možné jen pokud porovnáváme jedinou vlastnost. Většina skutečných obrazů je však o poznání komplexnější, než abychom mohli vybrat právě celý objekt porovnáním pouze jedné vlastnosti, například jasu. Opět můžeme porovnávat kterýkoliv barevný kanál, hodnoty jasu či nějaké statistické vlastnosti, například průměr hodnot vybrané vlastnosti v okolí porovnávaného pixelu. Vybereme-li dvojici

takovýchto vlastností a pokusíme-li se porovnat tyto dvojice v některých dvou pixelech, zjistíme důležitý fakt. Některé dvojice vůbec nejde porovnávat, neboť pro uspořádanou dvojici požadujeme stejné uspořádání, jako bylo uvedeno v definici 2.3.

Tím však vyvstává problém, jak posoudit spjatost dvou pixelů, jež jsou si velmi podobné v první vlastnosti, ale velice málo ve vlastnosti druhé. Budou pak dva pixely, kde jeden má vysokou hodnotu první vlastnosti a nízkou hodnotu druhé a ten druhý právě naopak, patřit k jednomu fuzzy objektu? Mohli bychom se pokusit dvojici vlastností nějak násilně linearizovat, například porovnávat jejich součty, součiny či vytvořit nějakou funkci závislou na obou rozdílech. Taková metoda by ale mohla způsobovat velké chyby a v žádném případě by dostatečně dobře nevystihla obraz, neboť různé vlastnosti se mohou značně měnit i při zachování podobnosti. To pak ovšem není metoda ani příliš užitečná, ani elegantní.

Jako zobecnění lineární metody se nabízí vytvoření aparátu, jež bude porovnávat celý tento soubor vlastností. Proto nyní afinitu nahradíme *multikritériem* (nadále jen *kritérium*), které seskupí afinity pro různé vlastnosti, takže jeho hodnoty budou tvořit částečně uspořádanou množinu.

**Definice 3.6.** *Zobrazení $\xi : V^2 \to P$, kde $(P, \leq)$ je částečně uspořádaná množina, splňující $\forall x, y, z \in V$:*

- *$\xi$ je symetrické, tj. $\xi(x, y) = \xi(y, x)$*
- *$\xi$ má maximum na diagonále, tedy $\xi(x, x) \geq \xi(y, z)$*

*nazveme* kritérium. *[6, str. 340]*

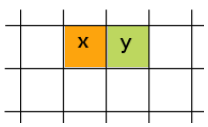
Druhá podmínka značí, že pixel je sám sobě nejpodobnější, což zajistí existenci největšího prvku $\top$ v oboru hodnot $\xi$. Platí navíc $\forall x \in V\ \xi(x, x) = \top$, tedy že takový největší prvek existuje pro kterýkoliv pixel. Největší prvek lze získat z každého pixelu, platí $\forall x, y, \in V$ je $\xi(x, x) = \xi(y, y)$.

Částečně uspořádaná množina $P$ je tvořena všemi možnými hodnotami kritéria, které lze nad daným obrazem vytvořit.

Zobrazení $\xi : V^2 \to P$ je *surjektivní*, jestliže každý prvek $P$ má nějaký vzor ve $V^2$, tedy jeho obraz $Im(\xi) = P$. Zahrneme-li tedy požadavek surjektivity $\xi$, bude to znamenat pouze to, že uspořádaná množina $P$ obsahuje pouze prvky takto vytvořené a žádné navíc. Proto může být $\top$ i největším prvkem uspořádané množiny $P$.

Prakticky, pokud budeme porovnávat například dvě vlastnosti jako jsou rozdíl hodnot v červeném kanálu a rozdíl v modrém kanálu, bude největší prvek $\top$ vypadat jako dvojice $\xi(x, x) = (0, 0)$.

Pro dva rozdílné pixely bude kritérium vracet poněkud pestřejší hodnoty. Můžeme se rozhodnout přidat ještě jednotlivé kritérium vzdálenosti, s tím, že toto budeme vždy uvádět jako první. Vezměme si dva pixely, například $x$ s hodnotami RGB $f(x) = [253, 164, 13]$ a $y$ s hodnotami $f(y) = [189, 215, 95]$, které leží vedle sebe, jak jsou znázorněny na následujícím obrázku. Kritérium bude vypadat jako $\xi(x, y) = (1, 64, 82)$



Vraťme se nyní k *relaci spjatosti* $\Phi$, $\Phi(x,y,z) \Leftrightarrow z \in \Omega(x,y)$, neboť podrobným rozebráním jejích vlastností získáme základ celé metody. V pojmech fuzzy relací dostáváme následující

$$\Phi(x,y,z) \;\Leftrightarrow\; \mu(x,z) \geq \mu(x,y) \;\wedge\; \mu(y,z) \geq \mu(x,y)\,.$$

Další analýzou této relace zjistíme následující fakt.

**Věta 3.1.** *Pro daná* $x,\ y,\ z \in V$ *je relace* $\Phi(x,y,z)$ *ekvivalentní tvrzení:*
$\forall\langle x_0,\ x_1, \cdots, x_n\rangle,\ x_0 = x,\ x_n = y \quad \exists\langle y_0,\ y_1, \cdots, y_k\rangle,\ y_0 = x,\ y_k = z$ *takový, že* $\forall\,(y_{i-1}, y_i)$
$\exists\,(x_{j-1}, x_j)$ *splňující* $\psi(x_{j-1}, x_j) \leq \psi(y_{i-1}, y_i)\,.$ *[6, str. 341]*

Tato věta v podstatě říká, že jakkoliv se dostaneme z bodu $x$ do bodu $y$, vždy na cestě existuje nějaký článek s afinitou stejnou či horší, než je podél nějaké cesty z $x$ do $z$. Neboli na každé cestě do $y$, tedy i na té nejlepší, je alespoň jeden článek s horší afinitou, než je na každém článku alespoň jedné cesty do $z$. Jedná se výchozí tvrzení pro další rozšiřování metody. Důkaz vyplývá z důkladné analýzy předchozího vztahu pro relaci spjatosti a je dohledatelný v [6].

Proto pro každý bod, který náleží fuzzy objektu stačí najít cestu, kde je každý článek lepší, než je alespoň nějaký článek té nelepší cesty do bodu $y$. Z čehož ale neplyne, že by musely všechny články mít navzájem porovnatelnou afinitu, čímž se jednoduše ztrácí požadavek lineárního uspořádání.

Bez tohoto omezení nyní můžeme afinitu nahradit nelineárním kritériem $\xi : V^2 \to (P, \leq)$ splňujícím oba dodatečné požadavky. Dostaneme tak nelineární relaci fuzzy spjatosti.

**Definice 3.7.** *Ternární relaci* spjatosti *definujeme pro pixely* $x, y, z \in V$ *jako*
$x, y, z \in \Phi \;\Leftrightarrow$
$\forall\langle x_0,\ x_1, \cdots, x_n\rangle,\ x_0 = x,\ x_n = y \quad \exists\langle y_0,\ y_1, \cdots, y_k\rangle,\ y_0 = x,\ y_k = z$ *, že*
$\forall\,(y_{i-1}, y_i)\ \ \exists\,(x_{j-1}, x_j)\ \ $ *splňující* $\ \xi(x_{j-1}, x_j) \leq \xi(y_{i-1}, y_i)$ *[6, str. 341]*

Již zde je jasné, že oproti lineární metodě zde bude třeba složitější prostředek pro porovnávání kritérií. Proto si před další specifikací metody vybudujeme nejdříve jistý teoretický základ. Zadefinujeme si některé algebraické struktury, které nám lépe pomohou popsat vlastnosti hledaného objektu.

## 3.3. Svazy

Stěžejní vlastností je možnost porovnání hodnot kritéria s prahem. Ovšem zde narážíme na problém. I pokud bude prahem nějaká hodnota z uspořádané množiny $P$, může být obtížné říci, zda je některé kritérium lepší, než tento práh. Jednoduše takto nelze posuzovat dva nesrovnatelné prvky. Bylo by však dobré, kdybychom o nesrovnatelných prvcích mohli říci alespoň, zda jsou oba větší, než nějaký třetí prvek, jejich infimum. Ani takové porovnání ale nemusí být v uspořádané množině možné, proto abychom takovou vlastnost zajistili, zavedeme zvláštní typ uspořádané množiny, svaz.

**Definice 3.8.** *Uspořádaná množina* $(L, \leq)$ *se nazývá* svaz *právě tehdy, když pro každou dvojici* $x, y \in L$ *existuje* $\bigvee\{x, y\}$ *a* $\bigwedge\{x, y\}$. *[4, str. 3]*

Neboli ve svazu mají každé dva prvky supremum a infimum. Není ale nutné omezovat se pouze na dvojice. Dá se totiž dokázat, že ve svazu má supremum i infimum každá konečná neprázdná podmnožina $H \subseteq L$.

Pro jednoprvkovou množinu je supremum i infimum rovno tomuto prvku. Má-li $H$ dva prvky, dostáváme předchozí případ. A pokud je $H = \{x, y, z\}$ tříprvková, je supremum $\bigvee\{\bigvee\{x, y\}, z\}$ a infimum $\bigwedge\{\bigwedge\{x, y\}, z\}$. Takto postupně lze získat supremum a infimum libovolné $n$-prvkové podmnožiny svazu $L$.

Supremum a infimum pro dva prvky lze dle [4, str. 4] značit i následovně

$$\bigvee\{x, y\} = x \vee y$$
$$\bigwedge\{x, y\} = x \wedge y,$$

čímž dostáváme dvě binární operace na $L$, neboli zobrazení typu $\omega : L^2 \to L$. Tyto operace budeme nazývat *spojení* pro $\vee$ a *průsek* pro $\wedge$. Svaz $L$ je navíc vůči oběma těmto operacím uzavřený, to znamená, že žádnou jejich aplikací nelze dostat prvek mimo $L$. Obě navíc splňují tyto čtyři pro svazy zásadní vlastnosti.

| | |
|---|---|
| Idempotence: | $x \vee x = x, \quad x \wedge x = x$ |
| Komutativita: | $x \vee y = y \vee x, \quad x \wedge y = y \wedge x$ |
| Asociativita: | $(x \vee y) \vee z = x \vee (y \vee z)$ |
| | $(x \wedge y) \wedge z = x \wedge (y \wedge z)$ |
| Absorpce: | $x \wedge (x \vee y) = x, \ x \vee (x \wedge y) = x$ |

Jak dál však tyto dvě operace souvisí s relací porovnání? To lze snadno ukázat. Pokud máme dva srovnatelné prvky, například $x, y \in L$ takové, že $x \leq y$, pak je zřejmé, že $\bigvee\{x, y\} = y$ a $\bigwedge\{x, y\} = x$ a proto taky $x \vee y = y$ a $x \wedge y = x$. Zároveň platí i opačná úvaha, proto

$$x \leq y \ \Leftrightarrow \ x \wedge y = x$$
$$x \leq y \ \Leftrightarrow \ x \vee y = y$$

Svaz pak místo relace porovnání můžeme úplně popsat pomocí těchto dvou operací, čímž získáme tzv. univerzální algebru $(L, \wedge, \vee)$ typu $(2, 2)$.

**Definice 3.9.** *Buď $A$ množina a $I$ množina indexů. Nechť $\forall i \in I$ je $\omega_i$ $n_i$-nární operace na $A$, tedy $\omega_i : A^{n_i} \to A$, $n_i \in \mathbb{N}_0$. Pak $\mathfrak{A} = \big(A, (\omega_i)_{i\in I}\big)$ nazveme* univerzální algebra s nosnou množinou $A$ *a souborem operací $(\omega_i)_{i\in I}$.*
*Soubor* arit $(n_i)_{i\in I}$ *nazveme* typ *algebry $\big(A, (\omega_i)_{i\in I}\big)$.* *[7, str. 4]*

**Definice 3.10.** *Algebra $\mathfrak{L} = (L, \wedge, \vee)$ tvoří* svaz *právě tehdy, když $L$ je neprázdná množina, $\wedge$ a $\vee$ jsou binární operace na $L$, jež jsou idempotentní, komutativní, asociativní a splňují vlastnosti absorpce.* *[4, str. 5]*

Často se však svaz označuje pouze svou nosnou množinou, tedy pouze $L$ namísto $\mathfrak{L}$.

**Definice 3.11.** *Nechť $L$ je svaz a neprázdná podmnožina $M \subseteq L$. Pak $M$ je* podsvaz *svazu $L$, když*

$$x, y \in M \ \Rightarrow \ x \wedge y \in M, \ x \vee y \in M$$

*[3, str. 41]*

Významnými podsvazy našeho svazu $L$ jsou ideál a filtr.

**Definice 3.12.** *Nechť $L$ je svaz.*

1. *Neprázdná podmnožina $I \subseteq L$ se nazývá* ideál*, pokud platí*
   - $x, y \in I \Rightarrow x \vee y \in I$
   - $x \in L,\ y \in I\ a\ x \leq y \Rightarrow x \in I$
2. *Neprázdná podmnožina $F \subseteq L$ se nazývá* filtr*, pokud platí*
   - $x, y \in F \Rightarrow x \wedge y \in F$
   - $x \in L,\ y \in F\ a\ x \geq y \Rightarrow x \in F$

*[3, str. 44]*

Ideál a filtr jsou vzájemně duální pojmy, které úzce souvisí s horními a dolními množinami. Ovšem v této práci budeme klást větší důraz především na filtry. Zjevnou analogií je fakt, že s daným prvkem patří do ideálu i všechny menší a do filtru i všechny větší. Jednoduše filtr je neprázdná dolní množina uzavřená vůči spojení a naopak filtr je neprázdná horní množina uzavřená vůči průseku. Ideál či filtr se nazývají *vlastní*, tvoří-li vlastní podmnožiny, neboli $I \subsetneqq L$, respektive $F \subsetneqq L$. Je zřejmé, že vlastní filtr nesmí obsahovat nejmenší prvek $\bot$. Pro každý prvek svazu $\forall x \in L$ horní množina generovaná jedním prvkem $\uparrow x$ je pokaždé filtrem, neboť ve svazu je na takové podmnožině uzavřenost vůči průseku automaticky splněna. Filtry či ideály generované jedním prvkem se nazývají *hlavní*.

## 3.4. Zobrazení mezi algebraickými strukturami

Zobrazením mezi algebraickými strukturami nám pomáhají přenášet některé teoretické výsledky z jedné množiny na druhou. Některá zobrazení totiž pomáhají zachovávat některé struktury. Zcela zásadním typem na uspořádaných množinách je *uspořádání zachovávající* zobrazení (nazývané též *monotonní* nebo *isotonní*).

**Definice 3.13.** *Mějme dvě uspořádané množiny $(P_1, \leq_1)$ a $(P_2, \leq_2)$. $\varphi : P_1 \to P_2$ se nazývá* isotonní*, jestliže*

$$x \leq y \ \Rightarrow\ \varphi(x) \leq \varphi(y).$$

*[3, str. 23]*

Pravým opakem je zobrazení *převracející uspořádání*, také nazývané *antitonní*, kde $x \leq y \ \Rightarrow\ \varphi(y) \leq \varphi(x)$.

Budeme-li nahlížet na svazy jako na algebry, nejdůležitějšími zobrazeními budou ta, která zachovávají obě operace.

**Definice 3.14.** *Nechť $L_1$ a $L_2$ jsou svazy. Zobrazení $\varphi : L_1 \to L_2$ je homomorfismus, jestliže $\varphi$ zachovává operace průseku a spojení, tj. $\forall x, y \in L_1$*

$$\varphi(x \vee_1 y) = \varphi(x) \vee_2 \varphi(y)$$
$$\varphi(x \wedge_1 y) = \varphi(x) \wedge_2 \varphi(y)$$

*[3, str. 43]*

Jelikož je svaz zároveň uspořádanou množinou, jsou svazové isomorfismy vždy isotonní. Dalším důležitým zobrazením mezi svazy je *isomorfismus*. Jedná se o homomorfismus, který je navíc bijektivní. Pokud existuje isomorfismus mezi svazy $L_1$ a $L_2$, pak je nazveme *isomorfní* a píšeme $L_1 \cong L_2$. Takové svazy můžeme v podstatě považovat za stejné a rozdíly mezi nimi můžeme zredukovat pouze na záležitost vhodného přeznačení. Pak je i možné nadále vynechat indexaci jednotlivých relací.

## 3.5. Vlastnosti svazů

Svaz jsme si již definovali dvěma různými způsoby, jako uspořádanou množinu a jako algebru. Je však skutečně nutné studovat různé definice téže struktury? Samozřejmě, neboť každý pohled nám může odhalit její různé vlastnosti. Například úplnost svazu nelze definovat algebraicky, neboť se jedná o pojem náležející čistě uspořádaným množinám.

**Definice 3.15.** *Svaz $L$ se nazývá* úplný*, pokud pro každou podmnožinu $H \subseteq L$ existuje $\bigwedge H$ a $\bigvee H$.* *[4, str. 24]*

Tato definice je v podstatě samoduální. Neboť existuje-li například supremum $x$ nějaké množiny, pak množina jejích horních závor má naopak infimum, opět prvek $x$. Proto bychom si v definici vystačili i pouze s jedním z obou pojmů. Navíc je zřejmé, že pokud každá podmnožina má supremum i infimum, musí i celý svaz mít nejmenší a největší prvek.

Podobně jako prvočísla v oboru přirozených čísel, ve svazech je podobně významná skupina nerozložitelných prvků.

**Definice 3.16.** *Nechť $L$ je svaz. Prvek $x \in L$ se nazývá* spojově ireducibilní*, jestliže platí*

- *pokud $L$ má nejmenší prvek, pak $x \neq \bot$*
- *$\forall u, v \in L$ platí $x = u \vee v \;\Rightarrow\; x = u$ nebo $x = v$.*

*Prvek $x \in L$ se nazývá* průsekově ireducibilní*, jestliže platí*

- *pokud $L$ má největší prvek, pak $x \neq \top$*
- *$\forall u, v \in L$ platí $x = u \wedge v \;\Rightarrow\; x = u$ nebo $x = v$.*

*[3, str. 53]*

V zájmu větší názornosti lze obě druhé podmínky definice přepsat pro spojově ireducibilní jako

$$\forall u, v \in L \quad u < x \text{ a } v < x \Rightarrow u \vee v < x$$

a pro průsekově ireducibilní platí vztah duální.

Vlastně se jedná o prvky, jež nejdou rozložit na průsek, případně spojení, jiných prvků, takové základní stavební kameny svazu. Množinu všech spojově ireducibilních prvků ve svazu $L$ označíme jako $\mathcal{J}(L)$ a množinu všech průsekově ireducibilních jako $\mathcal{M}(L)$. Obě tyto množiny jsou uspořádané stejně jako svaz $L$, neboť přebírají jeho uspořádání.

Jelikož jsme tyto prvky nazvali jakýmisi stavebními kameny svazu, je záhodno uvést, co tímto myslíme. Chtěli bychom vyjádřit každý prvek svazu jako supremum několika spojově ireducibilních prvků, případně jako infimum prvků průsekově ireducibilních. Obecně to sice provést nelze, ale alespoň na konečných svazech ano, jak popisuje následující věta.

**Věta 3.2.** *Nechť $L$ je konečný svaz. Pak $\forall x \in L$ existuje podmnožina $F \subseteq \mathcal{J}(L)$ taková, že $x = \bigvee F$.*

A totéž platí duálně, $\forall x \in L$ existuje podmnožina $F \subseteq \mathcal{M}(L)$ taková, že $x = \bigwedge F$.

Kromě vlastností, pomocí nichž jsme definovali svaz jako algebru, platí i další identity svazující algebraický koncept svazu a koncept svazu jako uspořádané množiny. Například každý svaz splňuje některé nerovnosti.

**Věta 3.3.** *Nechť $L$ je svaz a $x, y, z \in L$. Pak platí*

- $x \wedge (y \vee z) \geq (x \wedge y) \vee (x \wedge z)$
- $x \vee (y \wedge z) \leq (x \vee y) \wedge (x \vee z)$

*[3, str. 85]*

Tyto nerovnosti se nazývají *distributivní*. Nejedná se o jediné nerovnosti, jež jsou ve svazu splněny, ale tyto nás zajímají nejvíce, neboť dávají vzniknout jedné z důležitých tříd svazů. Takové, kde jsou splněny jako rovnost.

**Definice 3.17.** *Nechť $L$ je svaz. Tento svaz se nazývá* distributivní*, pokud splňuje* distributivní zákony*, $\forall x, y, z \in L$*

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$$

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$$

*[3, str. 86]*

## 3.6. Distributivní svaz generovaný uspořádanou množinou

Nyní se pokusíme propojit uspořádané množiny, jež jsme probrali v úvodu, a podstatně složitější strukturu distributivních svazů. Naším hlavním cílem je vystavět distributivní svaz, který jako podmnožinu obsahuje právě uspořádanou množinu $P$, která popisovala všechny hodnoty kritéria v obraze. Svaz by pak mohl popisovat jejich vzájemné vztahy.

Pak by ale jistě bylo vhodné, abychom takový svaz dokázali určit jednoznačně. K tomu nám dopomůže fakt, že každý prvek konečného svazu lze zapsat pomocí ireducibilních prvků. V našem případě lze konečnost uspořádané množiny $P$ považovat za splněnou, neboť v praxi lze pracovat pouze s konečnými obrazy, proto lze získat zase jen konečný počet hodnot kritéria. Konečnost svazu bude souviset právě s konečností množiny $P$. Zároveň zde ukážeme, jak distributivní svazy souvisí s horními množinami.

Mějme uspořádanou množinu $P$. Množinu všech horních množin množiny $P$ označíme $\mathscr{U}(P)$. Jelikož se vlastně jedná o soustavu jistých podmnožin $P$, můžeme na $\mathscr{U}(P)$ zavést

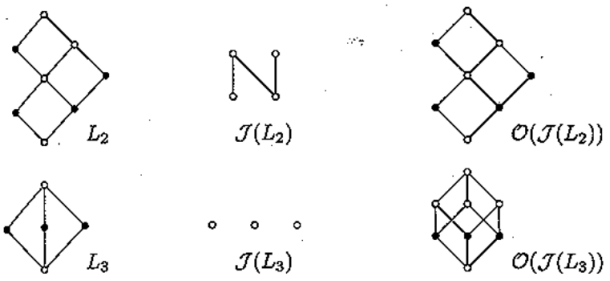
uspořádání podle množinové inkluze $\subseteq$, čímž získáme další uspořádanou množinu. Navíc průnikem či sjednocením horních množin získáme opět horní množinu. Symbolicky, pokud $\{A_i\}_{i\in I} \subseteq \mathscr{U}(P)$, pak $\bigcap_{i\in I} A_i \in \mathscr{U}(P)$ a také $\bigcup_{i\in I} A_i \in \mathscr{U}(P)$. Takže $\mathscr{U}(P)$ tvoří úplný svaz, jenž nazveme *svaz horních množin množiny* $P$. Ovšem dojde zde k jistému jevu. Pokud máme v uspořádané množině $P$ dva prvky $x, y$, kde $x \leq y$, pak $\uparrow y \subseteq \uparrow x$, neboť zkrátka ten menší prvek má „nad sebou“ více dalších prvků. Proto také v $\mathscr{U}(P)$ bude uspořádání opačné oproti $P$. O tomto pojednává [3, str. 37].

Nyní si ukážeme, že $\forall x \in P$ tvoří $\uparrow x$ spojově ireducibilní prvek svazu $\mathscr{U}(P)$. Předpokládejme, že $\uparrow x = U \cup V$, kde $U, V \in \mathscr{U}(P)$, zde je zřejmé, že $x$ patří do jedné z těchto množin, proto lze bez újmy na obecnosti předpokládat $x \in U$. Pak $\uparrow x \subseteq U$, ale zároveň $\uparrow x = U \cup V$, proto $\uparrow x \supseteq U$, tudíž musí platit $\uparrow x = U$. Čímž jsme si ověřili $\uparrow x \in \mathcal{J}(\mathscr{U}(P))$, jak je uvedeno v [3, str. 116].

Dále jelikož je $P$ konečná, tak má i konečný počet horních množin $\uparrow x$ generovaných jedním prvkem. Každá další horní množina $U \in \mathscr{U}(P)$ je sjednocením několika horních množin $\{\uparrow x_j\}_{j\in J\subseteq I}$. Proto jednak $\mathscr{U}(P)$ bude konečná a navíc všechny $\uparrow x$ budou tvořit právě množinu spojově ireducibilních prvků $\{\uparrow x \mid x \in P\} = \mathcal{J}(\mathscr{U}(P))$.

**Věta 3.4.** *Nechť je $P$ konečná uspořádaná množina. Pak zobrazení $\varepsilon : x \mapsto \uparrow x$ je antitonní zobrazení $(P, \leq)$ na $(\mathcal{J}(\mathscr{U}(P)), \supseteq)$.* *[3, str. 116]*

Na příkladech níže je uveden duální případ ke svazu dolních množin, který je však striktně analogický. Můžeme si na nich povšimnout zajímavého důsledku. Nezávisle na konkrétních vlastnostech výchozí uspořádané množiny, tvoří množina jejích dolních množin distributivní svaz. Tato vlastnost souvisí s následující větou, neboť množinou ireducibilních prvků svazu může být prakticky jakákoliv uspořádaná množina.




Obrázek 3.1: *Ukázka svazu dolních množin nad spojově ireducibilními prvky distributivního (nahoře) a nedistributivního (dole) svazu. $\mathcal{O}$ značí svaz dolních množin nad uspořádanou množinou $\mathcal{J}(L_i)$ [3, str. 117]*

**Věta 3.5** (Birkhoffův reprezentační teorém)**.** *Nechť $L$ je konečný distributivní svaz. Pak zobrazení $\delta : L \to \mathscr{U}(\mathcal{J}(L))$ definované jako*

$$\delta(x) = \{y \in \mathcal{J}(L) \mid x \leq y\}$$

*je isomorfismus $L$ na $\mathscr{U}(\mathcal{J}(L))$.* *[3, str. 118]*

Ovšem pokud svaz $L$ nebude distributivní, tvrdíme, že tímto postupem nad ním takový vytvoříme. Tak jako tak jsme dokázali svaz vystavět z ireducibilních prvků, jak bylo naznačeno dříve.

**Tvrzení 3.1.** *Nechť $P$ je uspořádaná množina. Svaz $\mathscr{U}(P)$ je distributivní.*

Zde si nastíníme ideu důkazu. Horní množina je podmnožinou výchozí množiny, proto se svaz $\mathscr{U}(P)$ skládá z podmnožin $P$. Mějme libovolné horní množiny $A, B, C \subseteq P$. Tvrdíme tedy, že

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

Sjednocením i průnikem horních množin je opět horní množina a uvedená identita na systému podmnožin $\mathfrak{P}(P)$ platí.

Navíc, když bude mít uspořádaná množina $P$ největší prvek, bude i systém všech neprázdných horních množin $\mathscr{U}^*(P)$ tvořit distributivní svaz. Protože má-li $P$ největší prvek, pak ho obsahuje každá horní množina, která není generována prázdnou množinou. Proto průnik prázdných množin nebude nikdy prázdný, tudíž bude prvkem $\mathscr{U}^*(P)$. Navíc největší prvek $P$ bude tvořit nejmenší prvek v $\mathscr{U}^*(P)$.

### 3.6.1. Volné distributivní svazy

Získali jsme sice distributivní svaz $\mathscr{U}(P)$, jehož prvky tvoří všechna možná kritéria a jejich supréma, to nám ale zatím nestačí, neboť pro popis objektů v obraze se stále jedná o příliš chudý aparát. Abychom mohli zachytit všechny možné vztahy kritérií, byl by vhodnější svaz volný.

*Identita* (rovnost) ve svazu je výraz tvaru $p = q$, kde $p$ a $q$ jsou jsou polynomy. Polynom nad svazem $L$ je výraz $n$ neurčitých $x_1, \cdots, x_n$, které mohou nabývat hodnot z $L$ a jsou spojeny operacemi $\wedge$, $\vee$ a závorkami. Polynom sám nabývá také hodnot z $L$.

**Definice 3.18.** *Řekneme, že* identita $p = q$ platí ve svazu $L$ *právě tehdy, když* $p(x_1, \cdots, x_n) = q(x_1, \cdots, x_n)$, $\forall x_1, \cdots, x_n \in L$. *[4, str. 28]*

**Definice 3.19.** *Nechť $L$ je svaz. Pro každou podmnožinu $H \subseteq L$, $H \neq \emptyset$ existuje nejmenší podmnožina $\langle H \rangle \subseteq L$ obsahující $H$ a uzavřená vůči $\wedge$ a $\vee$. Tedy $\langle H \rangle$ je podsvazem $L$, který nazýváme* podsvaz $L$ generovaný $H$ *a $H$ je* generující množina *svazu $\langle H \rangle$. [4, str. 17]*

Volný svaz je nejobecnější ze svazů. V našem případě ale budeme požadovat navíc jeho distributivitu. Prakticky se jedná o nejobecnější distributivní svaz, který lze vytvořit nad nějakou vstupní množinou $P$, jejíž prvky navíc mohou splňovat nějaká uspořádání. Nyní si tedy řekneme, co je volnost vzhledem k nějaké třídě svazů.

**Definice 3.20.** *Nechť jsou $p_i = q_i$ identity $\forall i \in I$. Třída $\mathbf{K}$ všech svazů splňujících všechny identity $p_i = q_i$, $i \in I$, se nazývá* ekvacionální třída svazů. *Ekvacionální třída je* triviální, *obsahuje-li jen jednoprvkové svazy. [4, str. 32]*

Například třída distributivních svazů $\mathbf{D}$ kromě identit idempotentních, komutativních, asociativních a absorpčních, aby vůbec prvky tvořily svaz, musí splňovat i identity distributivní. Nyní si ukážeme, co znamená pojem volnosti vzhledem ke generující množině $P$, která splňuje nějaké nerovnosti, pročež můžeme předpokládat, že tvoří uspořádanou množinu.

**Definice 3.21.** *Nechť $P$ je uspořádaná množina a $\mathbf{K}$ je ekvacionální třída svazů. Svaz $F_{\mathbf{K}}(P)$ nazýváme* volný nad $\mathbf{K}$ generovaný uspořádanou množinou $P$ *právě tehdy, když jsou splněny následující podmínky*

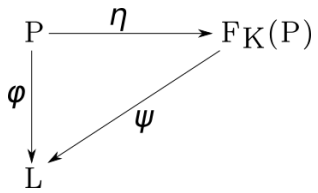
- $F_{\mathbf{K}}(P) \in \mathbf{K}$
- $P \subseteq F_{\mathbf{K}}(P)$ *a pro* $\forall x, y, z \in P$ *platí:*

  $\bigwedge\{x, y\} = z \ v \ P \Leftrightarrow \ x \wedge y = z \ v \ F_{\mathbf{K}}(P)$ *a* $\bigvee\{x, y\} = z \ v \ P \Leftrightarrow \ x \vee y = z \ v \ F_{\mathbf{K}}(P)$
- $\langle P \rangle = F_{\mathbf{K}}(P)$*, tj.* $F_{\mathbf{K}}(P)$ *je nejmenší svaz dané třídy obsahující* $P$
- *Nechť* $L \in \mathbf{K}$ *je libovolný svaz dané třídy a mějme isotonní zobrazení* $\varphi : P \to L$ *s vlastností* $\bigwedge\{x, y\} = z \ v \ P \Rightarrow \ \varphi(x) \wedge \varphi(y) = \varphi(z)$ *a* $\bigvee\{x, y\} = z \ v \ P \Rightarrow \varphi(x) \vee \varphi(y) = \varphi(z)$.

  *Pak existuje svazový homomorfismus* $\psi : F_{\mathbf{K}}(P) \to L$*, který rozšiřuje* $\varphi$*, tj.* $\forall x \in P$ *je* $\varphi(x) = \psi(x)$.

*[4, str. 32]*

Tato definice je poněkud nepřehledná a neintuitivní, pokusíme se ale alespoň nastínit význam jednotlivých bodů. Poslední podmínku lze vyjádřit následujícím diagramem.




Obrázek 3.2: *Komutující diagram, [4, str. 33].*

Říkáme, že tento diagram komutuje, to znamená, že $\varphi = \psi \circ \eta$. Zobrazení $\varphi$ je inkluze uspořádané množiny do svazu $L$ se zachováním uspořádání a složení identity a homomorfismu $\psi$ zobrazí tytéž prvky přes svaz $F_{\mathbf{K}}(P)$ zase na odpovídající místa v $L$. Takže svaz $F_{\mathbf{K}}(P)$ je oproti $P$ obohacen o ty prvky, díky nimž spadá do dané kategorie.

Dá se ukázat, že volný svaz $F_{\mathbf{K}}(P)$ lze vytvořit právě tehdy, když existuje i svaz $L$ dané třídy s vlastností $\forall x, y, z \in P : \ \bigwedge\{x, y\} = z$ v $P \Leftrightarrow \ x \wedge y = z$ v $L$ a $\bigvee\{x, y\} = z$ v $P \Leftrightarrow \ x \vee y = z$ v $L$, [4, str. 34].

Takovýto svaz $L$ lze vytvořit pro každou uspořádanou množinu $P$, jak je uvedeno v [4, str. 36].

### 3.6.2. Dvojí vytvoření svazu horních množin

Volný svaz bude pro naše účely velmi vhodný, ale zatím nemusí být zcela zřejmé, proč tomu tak je, ani jak volný svaz vytvořit. Opět ale budeme chtít, aby bylo zaručeno, že tento svaz vytvoříme jednoznačně. Nabízí se použít stejný postup, jaký již byl jednou úspěšný při vytváření distributivního svazu, a to vytvoření soustavy horních množin.

První aplikací jsme z obecné uspořádané množiny $P$ získali distributivní svaz $\mathscr{U}(P)$ s inkluzí $\varepsilon : x \mapsto \uparrow x$. Přičemž jsme ukázali, že obrazy prvků z $P$ tvoří v tomto svazu spojově ireducibilní prvky. Nadále však budeme uvažovat pouze systém neprázdných horních množin $\mathscr{U}^*(P)$. Přestože v [4] ani [3] taková omezení nepožadují, v našem případě si vystačíme s neprázdnými podmnožinami, neboť dvojí inkluzí vytvoříme všechny potřebné

prvky. Ponecháním prázdné horní množiny by vznikly prvky navíc, nula a jednička svazu, které by nebyly shodné s $\bigwedge\{\eta(P)\}$ a $\bigvee\{\eta(P)\}$.

Když nyní nad $\mathscr{U}^*(P)$ vytvoříme opět množinu všech neprázdných horních množin, kterou označíme $\mathscr{U}^*(\mathscr{U}^*(P)) = \mathscr{W}(P)$, získáme nový distributivní svaz s uspořádáním $\leq$. Zobrazení

$$\eta = \varepsilon_{\mathscr{U}(P)} \circ \varepsilon_P, \quad \text{které} \quad \eta : x \mapsto \uparrow\uparrow x$$

je složením dvou antitonních zobrazení, čímž zase dostáváme zobrazení isotonní. Tedy $x \leq y$ v $P \Rightarrow \uparrow\uparrow x \leq \uparrow\uparrow y$ ve $\mathscr{W}(P)$.

V tomto případě, když je již $\mathscr{U}^*(P)$ svazem, lze v souvislosti s jeho horními množinami $\uparrow U,\ U \in \mathscr{U}^*(P)$, hovořit o hlavních filtrech. Tudíž platí, že hlavní filtry $\mathscr{U}^*(P)$ tvoří spojově ireducibilní prvky svazu $\mathscr{W}(P)$. Jenže některé tyto filtry jsou generovány horními množinami tvaru $\uparrow x$ svazu $P$. Zdůrazněme, že tyto hrály úlohu spojově ireducibilních prvků v $\mathscr{U}(P)$. Další ideály jsou pak tvořeny jako spojení těchto ireducibilních. Ovšem při druhém zobrazení $\varepsilon_{\mathscr{U}^*(P)} : \uparrow x \mapsto \uparrow\uparrow x$ dojde k dalšímu převrácení uspořádání, proto se význam těchto prvků změní.

Prvky, které v $\mathscr{U}^*(P)$ byly spojově ireducibilní, se stanou zároveň průsekově ireducibilními a prvky, jež byly spojeními ireducibilních prvků, se zobrazením do $\mathscr{W}(P)$ přemění ve průseky obrazů prvků $P$. Tudíž každý prvek ve $\mathscr{W}(P)$ lze napsat jako spojení prvků $\eta(P)$ či jejich průseků.

### Volný distributivní svaz nad P

Nyní zbývá pouze ověřit, jakou strukturu jsme získali dvojím vytvořením svazu horních množin.

**Věta 3.6.** *Mějme uspořádanou množinu $P$, pak $\mathscr{W}(P)$ je volný distributivní svaz generovaný množinou $P$.*

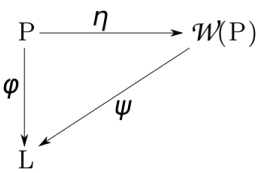
*Důkaz.* Stačí ověřit podmínky v definici 3.21, přičemž pracujeme s ekvacionální třídou distributivních svazů $\mathbf{D}$.

$\mathscr{W}(P) \in \mathbf{D}$, protože distributivita svazu horních množin plyne z tvrzení 3.1.

$P \subseteq \mathscr{W}(P)$ je zřejmé, dále $\eta$ je složením dvou antitonních zobrazení, tudíž je isotonní. Proto $\forall x, y \in P$: $x \leq y \Rightarrow \eta(x) \leq \eta(y) \Rightarrow \uparrow\uparrow x \leq \uparrow\uparrow y \Rightarrow \uparrow x \geq \uparrow y \Rightarrow x \leq y$, zachovají se tedy i případná suprema a infima.

$P$ generuje $\mathscr{W}(P)$, neboť jakoukoliv aplikací suprem a infim získáme všechny prvky z $\mathscr{W}(P)$ a se zachováním distributivity nelze získat žádné další.

Je nutné ukázat, že pro daný svaz $L \in \mathbf{D}$ a izotonní zobrazení $\varphi : P \to L$ existuje zobrazení $\psi : \mathscr{W}(P) \to L$ takové, že $\psi \circ \eta = \varphi$ je homomorfismus.




Mějme prvek $\alpha \in \mathscr{W}(P)$. Jak bude ukázáno později, lze jej jednoznačně zapsat (až na komutativitu a idempotenci) ve tvaru $\alpha = \bigvee_{i \in I} \bigwedge_{j \in J_i} a_j$, $a_j \in P$. Zobrazení $\psi$ lze definovat

jako $\psi(\alpha) = \bigvee_{i\in I} \bigwedge_{j\in J_i} \varphi(a_j)$, což ukazuje jeho existenci. Navíc je určeno jednoznačně až na komutativitu a idempotenci.

Dále $\forall x \in P:\ (\psi \circ \eta)(x) = \psi(\eta(x)) = \psi(\uparrow\uparrow x) = \varphi(x)$ a chceme ukázat, že se jedná o homomorfismus, $\varphi(\alpha \vee \beta) = \varphi\left(\left(\bigvee_{i\in I} \bigwedge_{j\in J_i} x_j\right) \vee \left(\bigvee_{k\in K} \bigwedge_{l\in J_k} x_l\right)\right) = \varphi\left(\bigvee_{i\in I\cup K} \bigwedge_{j\in J_i} x_j\right)$. Podobný vztah platí pro průseky, tedy $\varphi$ je homomorfismus.

Tímto jsme ukázali, že $\mathscr{W}(P)$ je volný nad třídou distributivních svazů generovaný množinou $P$. □

## 3.7. Horní množiny v obraze

Jednotlivé hodnoty kritéria jsou prvky uspořádané množiny $P$. Jelikož jsme již zavedli horní množiny nad uspořádanou množinou, nabízí se kritéria v relaci spjatosti přepsat do pojmů tohoto systému podmnožin $P$.

**Věta 3.7.** *Pro uspořádanou množinu $(P, \leq)$, množiny $A$, $B$ a zobrazení $\alpha : A \to P$ a $\beta : B \to P$ jsou následující tvrzení ekvivalentní*

- $\forall a \in A\ \exists b \in B$ *takové, že* $\beta(b) \leq \alpha(a)$
- $\uparrow\{\alpha(a) \mid a \in A\} \subseteq \uparrow\{\beta(b) \mid b \in B\}$

To, že první tvrzení implikuje druhé, je zřejmé, neboť již víme, že pokud je $\alpha(a)$ větší, pak generuje menší horní množinu. Druhý směr znamená, že pokud je jedna horní množina podmnožinou druhé, musí v té druhé být alespoň jeden menší, což je také poměrně zřejmé.

Důsledkem je, že můžeme upravit relaci spjatosti do následujícího tvaru.

**Věta 3.8.** *Nechť $x, y, z \in V$, pak $x, y, z \in \Phi \Leftrightarrow$*
$\uparrow\{\uparrow\{\xi(x_{i-1}, x_i) \mid 1 \leq i \leq n\} \mid \langle x_0, \cdots, x_n\rangle,\ x_0 = x,\ x_n = y\} \subseteq$
$\subseteq \uparrow\{\uparrow\{\xi(y_{j-1}, y_j) \mid 1 \leq j \leq m\} \mid \langle y_0, \cdots, y_m\rangle,\ y_0 = x,\ y_m = z\}$

*[6, str. 342]*

Důkaz této věty plyne z věty předchozí, kterou aplikujeme nejdříve na zobrazení $\xi$, které zobrazuje dvojici pixelů do uspořádané množiny $P$ a následně na zobrazení každé cesty mezi dvěma body do $\mathscr{W}(P)$.

### 3.7.1. Praktická ukázka

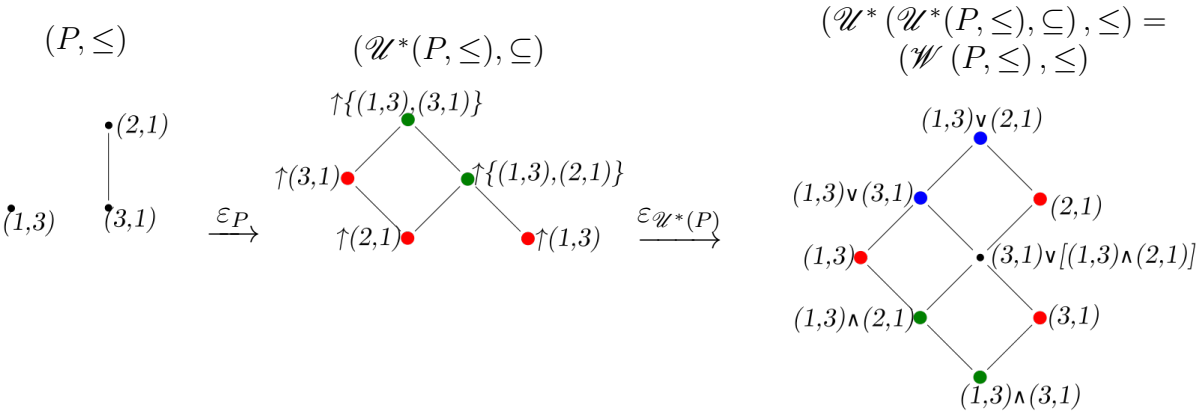
Pro větší názornost uvedeného postupu a lepší porozumění celému definovanému konceptu zde uvádíme jednoduchý příklad. Vlevo je znázorněna uspořádaná množina se třemi prvky $P = \{a, b, c\}$, z nichž dva jsou porovnatelné. Na obrázku 3.6 byl generován svaz horních množin (vpravo dole) nad tříprvkovou množinou bez uspořádání (uprostřed dole). Povšimněme si, že když musíme respektovat uspořádání, je svaz $\mathscr{U}(P, \leq)$ znatelně menší. Rozdíl by se prohloubil u svazu $\mathscr{W}(P, \leq)$.

Navíc jsme ve svazu $\mathscr{W}(P, \leq)$ znázornili ireducibilní prvky. Spojově ireducibilní jsou vyznačeny zeleně, ty jsou zároveň obrazy prvků $\mathscr{U}^*(P) \setminus \mathcal{J}(\mathscr{U}^*(P))$. Průsekově ireducibilní

jsou označeny modře a obrazy původních prvků, jenž jsou zároveň průsekově i spojově ireducibilní, jsou červené.

Pro větší názornost je svaz navíc opatřen některými skutečně použitelnými hodnotami dvouprvkového kritéria $\xi$.

Konkrétní postup vypadá následovně: například pro podmnožinu $\{(1,3),(3,1)\} \subset P$ je $\uparrow\{(1,3),(3,1)\} = \{(1,3),(2,1),(3,1)\}$. Takto vytvoříme horní množiny pro všechny neprázdné podmnožiny $P$, díky uspořádání si mohou být některé z nich rovny, například $\uparrow(3,1) = \uparrow\{(2,1),(3,1)\}$. Tyto podmnožiny pak uspořádáme dle množinové inkluze. Tentýž postup zopakujeme, čímž získáme množinu $\mathscr{W}(P)$. Druhá inkluze $\varepsilon_{\mathscr{U}^*(P)}$ funguje prakticky stejně, pouze máme větší systém podmnožin.




## 3.8. Kategoriální pohled

Ačkoliv jsme doposud pojem struktura používali spíše v intuitivním významu jakožto charakteristiku nějakého uspořádání či vlastností objektu, je kolem matematických struktur vybudována rozsáhlá teorie, teorie kategorií, vytvářející komplexní pohled na všechny typy objektů, jež se v celé matematice vyskytují. Tato teorie nám tedy může poskytnout poněkud odlišný náhled na svazy.

**Definice 3.22.** *$\mathscr{S}$ je* matematická struktura *(stručněji struktura), jestliže*

- *Pro každou množinu $X$ je určena množina $\mathscr{S}[X]$, jejíž prvky nazýváme* strukturace *množiny $X$ a dvojice $A = (X,\alpha)$, $\alpha \in \mathscr{S}[X]$, nazýváme* objekty *struktury $\mathscr{S}$.*
- *Pro každé dva objekty $A = (X,\alpha)$, $B = (Y,\beta)$ je určena množina zobrazení z $X$ do $Y$: $hom_{\mathscr{S}}(A,B) \subseteq Y^X$, jejíž prvky nazýváme* morfismy *z $A$ do $B$, přičemž platí:*
  1. *Pro libovolné dva morfismy $f : A \to B$ a $g : B \to C$ jejich složením vznikne morfismus $g \circ f : A \to C$.*
  2. *Pro každý objekt $A = (X,\alpha)$ je identické zobrazení $id_X : A \to A$ morfismem.*

*[1, str. 12]*

Příkladem matematické struktury může být právě struktura *svazů*, značená **Sva** nebo **Lat** (z anglického lattice). Struktura svazů není nic jiného než všechny uspořádané množiny, pro jejichž každé dva prvky existuje supremum a infimum. Její podstrukturou je struktura **DLat** distributivních svazů.

Objektem **Lat** je tedy svaz $A = (X, \leq)$ nebo třeba svaz $B = (Y, \preceq)$. Morfismy $hom_{\mathscr{S}}(A, B)$ jsou zobrazení $f : X \to Y$, jenž jsou svazovými homomorfismy, tedy že zachovávají suprema a infima, tj. $\forall x, y \in X$:

$$\begin{aligned} f(x \wedge_A y) &= f(x) \wedge_B f(y) \\ f(x \vee_A y) &= f(x) \vee_B f(y). \end{aligned}$$

Ovšem strukturou je například i struktura uspořádaných množin **Pos**, nebo třeba **Met** - struktura metrických prostorů, a řada dalších.

Především nám ale teorie kategorií může pomoci lépe porozumět pojmu volnosti svazu.

**Definice 3.23.** *Řekneme, že objekt $(X, \alpha)$ je* volný *nad množinou $M$, jestliže platí $M \subseteq X$ a pro každý objekt $(Y, \beta)$ a každé zobrazení $f_0 : Y \to M$ existuje právě jedno rozšíření na morfismus $f : (X, \alpha) \to (Y, \beta)$. To znamená, že $f$ je jediný morfismus takový, že $\forall m \in M : \ f(m) = f_0(m)$.* *[1, str. 59]*

Již na první pohled je jasné, že je tato definice o poznání jednodušší, než definice volného svazu uvedená dříve, takže by mohla být i poněkud srozumitelnější. V podstatě jediným zásadním požadavkem je to, že zobrazení $f_0$ musí být jednoznačně rozšiřitelné na zobrazení celé množiny $X$ na $Y$ s tím, že výsledné zobrazení musí být morfismem. To znamená, že musí zachovávat důležité operace či význačné prvky dané struktury. Intuitivně řečeno, pokud jsme ze zadaných prvků $M$ a pravidel pro morfismy na dané struktuře schopni jednoznačně „dopočítat“ zbylé obrazy prvků z $X$, pak je $(X, \alpha)$ volně vygenerován z $M$.

Většímu porozumění konceptu volnosti pomůže následující příklad. Struktura monoidů **Mon** je struktura množin s jednou asociativní binární operací a neutrálním prvkem. Tvrdíme, že objekt $(\mathbb{N}_0, +, 0)$ je volný nad množinou $M = \{1\}$. Morfismy na monoidech musí zachovat neutrální prvek a operaci, tedy $\forall x, y, \in X : \ f(x + y) = f(x) \cdot f(y)$. Za druhý objekt si zvolme třeba $(X, \cdot, 1) \in$ **Mon**. Zobrazení $f_0 : \{1\} \to X$ je dáno jako $f_0(1) = x$. Nadále postupujeme dle pravidel pro morfismy. Díky zachování neutrálního prvku je $f(0) = 1$ a kvůli zachování operace je $f(2) = f(1+1) = f(1) \cdot f(1) = x \cdot x = x^2$, $f(3) = x^3$ a tak dále, až vytvoříme obrazy všech prvků $\mathbb{N}_0$. Takže $(\mathbb{N}_0, +, 0)$ je zcela jistě volný nad $\{1\}$.

V našem případě je svaz $\mathscr{W}(P) \in$ **DLat** volný nad množinou $P$, neboť zobrazením $f_0$ je $\eta : P \to \mathscr{W}(P)$ a pokud vybereme libovolný svaz $L \in$ **DLat** a isotonní zobrazení $\varphi : P \to L$, pak existuje jediný svazový homomorfismus $\psi : \mathscr{W}(P) \to L$ takový, že $\varphi = \psi \circ \eta$. Tato podmínka je totiž v podstatě tatáž, jako poslední podmínka v předchozí definici volného svazu, již jsme už však ověřili. Není tak nutno ověřovat další tři podmínky.

Vytvořili jsme tedy vztah mezi uspořádanou množinou a distributivním svazem. Kategoricky vztahy mezi různými strukturami popisují funktory.

**Definice 3.24.** Funktor *z kategorie $\mathscr{K}$ do kategorie $\mathscr{L}$ je předpis $F : \mathscr{K} \to \mathscr{L}$, který*

- *každému objektu $A$ z kategorie $\mathscr{K}$ přiřadí objekt $F(A)$ z kategorie $\mathscr{L}$,*
- *každému morfismu $f : A \to B$ kategorie $\mathscr{K}$ přiřadí morfismus $F(f) : F(A) \to F(B)$ kategorie $\mathscr{L}$ tak, že platí:*

1. $F(g \circ f) = F(g) \circ F(f)$ *pro libovolné morfismy* $f : A \to B$ *a* $g : B \to C$ *v kategorii* $\mathscr{K}$
2. $F(id_A) = id_{F(A)}$ *pro každý objekt* $A \in \mathscr{K}$

*[1, str. 173]*

Takže navíc získáváme funktor z kategorie **Pos** do kategorie **DLat**, $\mathscr{W} : (P, \leq) \mapsto \mathscr{U}^*(\mathscr{U}^*(P, \leq), \subseteq)$. To znamená, že jakmile objevíme morfismus jedné uspořádané množiny na druhou, známe také morfismus mezi jejich volnými distributivními svazy.

Důležitým typem funktoru je *zapomínající funktor*, který je definován pro libovolnou strukturu $\mathscr{S}$. Definujeme jej předpisem

$$U_{\mathscr{S}} : \mathscr{S} \to \mathbf{Set}, \text{ který } \; U_{\mathscr{S}} : (X, \alpha) \mapsto X,$$

dle [1, str. 176]. Tento funktor v podstatě „zapomíná“ strukturaci objektu, to znamená, že mu přiřadí jeho nosnou množinu.

Jiná definice volné struktury uvedená v [1, str. 228] staví právě na volném funktoru. *Volný objekt nad množinou* $M$ je objekt $A_0$ takový, že platí $M \subseteq A_0$ a inkluze

$$v : M \to U(A_0)$$

má vlastnost, že pro každý objekt $A$ a každé zobrazení $f$,

$$f : M \to U(A)$$

existuje právě jeden morfismus $f^* : A_0 \to A$, který rozšiřuje zobrazení $f$. To znamená, že $U(f^*) \circ v \; = f$ je morfismus.

V našem případě to znamená, že $\mathscr{W}(P)$ volný nad množinou $P$, když platí $P \subseteq \mathscr{W}(P)$ a pro inkluzi

$$\eta : P \to U\left(\mathscr{W}(P)\right)$$

platí, že pro každý objekt (svaz) $L$ je každé zobrazení $f : P \to L$ jednoznačně rozšiřitelné na morfismus $f^* : \mathscr{W}(P) \to L$.

To také znamená, že $U(f^*) \circ \eta \; = f$ je morfismus. Čímž ale dostáváme obdobnou definici jako v předchozích případech. Stačí pouhé přeznačení $f = \varphi$ a $f^* = \psi$ a vidíme, že dostáváme poslední podmínku prví definice volného svazu, pouze s požadavkem jednoznačnosti navíc, ta je tam však splněna, vzhledem j jednoznačnosti toho, jak jsme $\psi$ definovali.

# 4. Termové prahování

Tato kapitola předkládá hlavní myšlenky včetně základních teoretických podkladů a metod pro vícekriteriální prahování obrazu. Hlavní myšlenky jsou nastoleny v [6], odsud v této kapitole přebíráme i většinu výsledků.

## 4.1. Spojitostní term

Prvky ve $\mathscr{W}(P)$ lze zapsat jen pomocí průseků a spojení prvků původní uspořádané množiny. U takového zápisu lze však dosáhnout jisté struktury, která pak bude výhodná především při zpracování výsledného algoritmu.

### Disjunktní normální tvar

Tento pojem používáme především pro popis termů v Booleově algebře. Term je ve zkratce souborem proměnných, které nabývají hodnot prvků algebry, spojených operacemi průseků, spojení a závorkami. Každý term booleovské algebry se dá převést na následující tvar.

**Definice 4.1.** *Řekneme, že booleovský term $p(x_1, \cdots, x_n)$ je v* disjunktním normálním tvaru *(DNF), jestliže je spojením konjunktivních klauzulí tvaru $x_1^{\varepsilon_1} \wedge \cdots \wedge x_n^{\varepsilon_n}$, kde $x_i^{\varepsilon_i}$ je rovno termu $x_i$ nebo jeho negaci. Termy tvaru $x_i^{\varepsilon_i}$ nazýváme* literály. *[3, str. 103]*

Přestože Booleova algebra je svaz jiných vlastností, než svazy, se kterými zde pracujeme, je dobře použitelný i v tomto kontextu. V podstatě se totiž nejedná o nic jiného, než popis struktury prvků ve svazu. Především v distributivních svazech nemáme definován pojem negace, ale již jsme zjistili, že prvek svazu $\mathscr{W}(P)$ je disjunkcí některých prvků, které mohou být zase zapsány jako konjunkce.

Tvary prvků ve $\mathscr{W}(P)$ jsou tedy jasnou analogií disjunktního normálního tvaru, přičemž literály tvoří právě prvky výchozí uspořádané množiny $P$, pouze bez negací.

Proto můžeme psát

$$\uparrow\{\uparrow\{\, x_j \mid j \in J_i\} \mid i \in I\} = \bigvee_{i \in I} \bigwedge_{j \in J_i} x_j.$$

Díky tomu, že prvky $\mathscr{W}(P)$ jsou jednoznačně určeny jako spojení ireducibilních prvků, které jsou opět jednoznačné vůči průsekům prvků původní množiny, je i zápis v disjunktním normálním tvaru vždy jednoznačný až na pořadí a opakování (což je však nepodstatné díky komutativitě a idempotenci). Toto je jedna ze zásadních výhod distributivních svazů.

Ještě si specifikujeme, jak v tomto zápisu fungují obě operace ve svazu $\mathscr{W}(P)$. Mějme dva prvky $\alpha, \beta \in \mathscr{W}(P)$ zapsané ve tvaru $\alpha = \bigvee_{i \in I} \bigwedge_{j \in J_i} a_j$ a $\beta = \bigvee_{k \in K} \bigwedge_{l \in J_k} a_l$. Pak platí

$$\alpha \vee \beta = \bigvee_{i \in I \cup K} \bigwedge_{j \in J_i} a_j \quad \text{a} \quad \alpha \wedge \beta = \bigvee_{(i,k) \in I \times K} \left( \bigwedge_{j \in J_i} a_j \wedge \bigwedge_{l \in J_k} a_l \right)$$

Opět se vraťme k relaci blízkosti. Operace s horními množinami, jaké jsme provedli ve větě 3.8, nápadně připomínají uvedený zápis. Pokusíme se toho využít a přepsat je na operace průseků a spojení nad jednotlivými kritérii. Takto získáme předpis

$$\uparrow\{\uparrow\{\xi(x_{i-1},x_i) \mid 1\le i\le n\} \mid \langle x_0=x, x_1, \cdots, x_n=y\rangle\} = \bigvee_{\langle x_0=x,x_1,\cdots,x_n=y\rangle} \bigwedge_{1\le i\le n} \xi(x_{i-1},x_i),$$

který bude zcela zásadní pro celou metodu.

**Definice 4.2.** *Pro daný pár pixelů $x, y$ v digitálním obraze $(V,\pi,f)$ s kritériem $\xi : V^2 \to (P,\le)$ je* spjatostní term pixelů $(x,y)$ *definován jako term ve $\mathscr{W}(P)$*

$$\kappa(x,y) = \bigvee_{\langle x_0=x,x_1,\cdots,x_n=y\rangle} \bigwedge_{1\le i\le n} \xi(x_{i-1},x_i).$$

Nadále můžeme pro zjednodušení značit $\mathscr{W}(P)$ jenom jako $L$. Takže spjatostní term bude zobrazení $\kappa : V^2 \to L$.

Nyní také lze přepsat celou relaci spjatosti do poměrně jednoduché nelineární podoby

$$(x,y,z)\in\Phi \;\Leftrightarrow\; \kappa(x,y)\le\kappa(x,z),$$

která je zcela analogická lineární relaci, pouze jsme $\mu$ nahradili za $\kappa$. Takto navíc můžeme přepsat i oba fuzzy objekty. Ovšem místo reálného prahu $t$ je nutné také použít nelineární obdobu. Jelikož $\kappa(x,y)$ je prvkem $L$, tak je nutné tento term porovnávat opět s jiným termem, proto jako práh použijeme $\tau\in L$.

$$\begin{aligned}\Theta(x,\tau) &= \{u\in V \mid \kappa(x,u)\le\tau\}\\ \Omega(x,y) &= \Theta\left(x,\kappa(x,y)\right) = \Theta\left(y,\kappa(x,y)\right).\end{aligned}$$

Tento postup je skutečně rozšířením původní metody, neboť pro lineárně uspořádanou množinu $P$ nejsou v obou přístupech rozdíly. Průsek na konečné lineárně uspořádané množině odpovídá minimu, spojení naopak maximu. Takže svazem $L$ bude opět řetězec, jako práh tudíž postačí reálná hodnota.

Ani význam obou fuzzy objektů se nijak významně nemění. Objekt $\Omega(x,y)$ je opět nejmenší objekt určený kritériem $\xi$, který obsahuje $x$ i $y$. Přestože je objekt $\Omega$ pro předem dané kritérium určen jednoznačně, ani v lineárním případě není právě triviální jej určit. Opět je nutné procházet celý obraz, aby vůbec bylo možné určit hodnotu fuzzy spojitosti $\kappa(x,y)$. Později zde představíme algoritmus, jak její hodnotu získat, ale brzy uvidíme, že oproti lineárnímu případu je mnohonásobně složitější.

Další analýzou relace spjatosti uvedené dříve a v definici 3.7 získáme následující ekvivalence

$$\begin{aligned}(x,y,z)\in\Phi \;&\Leftrightarrow\; \kappa(x,y) = \bigvee_{\langle x_0=x,x_1,\cdots,x_n=y\rangle} \bigwedge_{1\le i\le n} \xi(x_{i-1},x_i) \le \kappa(x,z) \;\Leftrightarrow\\ &\Leftrightarrow\; \forall\langle x=x_0,\, x_1,\cdots,x_n=y\rangle,\; \exists\langle x=y_0,\, y_1,\cdots,y_k=z\rangle :\\ &\qquad \forall\,(y_{i-1},y_i)\;\; \exists\,(x_{j-1},x_j)\;\; \xi\,(x_{j-1},x_j)\le\xi\,(y_{i-1},y_i)\end{aligned}$$

Pokud kritérium bere v úvahu také vzdálenost pixelů, nemusí být článek na cestě nutně tvořen přilehlými pixely, naopak dvojice pixelů s nějakou větší vzdáleností $d$ může být

výhodnější, pokud dává dobrou hodnotu dalších vlastností. Není proto nutné požadovat cestu přímo po článcích. Takže pro suprema a infima použijeme obecné indexové množiny $I$ a $J_i$, $i \in I$. $I$ tedy bude množina nějakých obecných cest z $x$ do $y$ s pohyby po nepřilehlých pixelech. Takže dostáváme

$$\bigvee_{i\in I}\bigwedge_{(u,v)\in J_i} \xi(u,v) \leq \kappa(x,z) \quad \Leftrightarrow \quad \forall i \in I\ \exists \langle x = y_0,\, y_1, \cdots, y_k = z\rangle :$$
$$\forall\,(y_{i-1}, y_i)\ \ \exists\,(u,v) \in J_i\ \xi\,(u,v) \leq \xi\,(y_{i-1}, y_i)$$

Pokud nyní místo kritérií použijeme literály $s \in P$ na cestách $J_i \subseteq P$, dostáváme

$$\bigvee_{i\in I}\bigwedge_{s\in J_i} s \leq \kappa(x,z) \Leftrightarrow \forall i \in I\ \exists \langle x = y_0,\, y_1, \cdots, y_k = z\rangle : \forall\,(y_{i-1}, y_i)\ \ \exists s \in J_i\ s \leq \xi\,(y_{i-1}, y_i)$$

Důsledkem je následující věta.

**Věta 4.1.** *Pro dané literály $s_j \in P \subseteq L$, $j \in \{1, \cdots, m\}$ a termy $\alpha_i \in L$ ve tvaru průseku literálů $s$ $i \in \{1, \cdots, n\}$, $m, n \in \mathbb{N}$, platí*

$$\bigwedge_{j\in\{1,\cdots,m\}} s_j \leq \kappa(x,z) \quad \Leftrightarrow$$
$$\Leftrightarrow \quad \exists \langle x = y_0,\, y_1, \cdots, y_k = z\rangle : \forall\,(y_{i-1}, y_i) \bigvee_{j=1}^{m} \left(s_j \leq \xi(y_{i-1}, y_i)\right),$$
$$\bigvee_{i\in\{1,\cdots,n\}} \alpha_i \leq \kappa(x,z) \quad \Leftrightarrow$$
$$\Leftrightarrow \quad \bigwedge_{i=1}^{n} \exists \langle x = y_0,\, y_1, \cdots, y_k = z\rangle : \forall\,(y_{i-1}, y_i)\,(\alpha_i \leq \xi(y_{i-1}, y_i)),$$

*kde operátory $\bigvee_{j=1}^{m}$ a $\bigwedge_{i=1}^{n}$ aplikované na formule značí logickou disjunkci a konjunkci.*

*[6, str. 343]*

Již víme, jaké jsou významy literálů, popisují spjatost dvojic pixelů. Vyjadřují v podstatě to, jak snadné či obtížné bylo přejít z jednoho pixelu na druhý. Ovšem přesunem ke složitější struktuře svazu se stává i význam termů poněkud složitějším. Uvedená věta nám dává náhled na to, jaký je v obraze význam termů.

Průsek literálů funguje v obraze spíše jako disjunkce. Při pohybu po cestě s termem ve formě konjunktivní klauzule, například $s_1 \wedge s_2$, se totiž můžeme pohybovat po pixelech spjatých s mírou $s_1$ *nebo* $s_2$, ale samotná cesta je popsána literály $s_1$ *a* $s_2$.

Pro lepší pochopení důvodu této zdánlivé nekonzistence se přesuňme zpět ke svazu $\mathscr{U}(P)$. Prvek, který byl v $L$ ve tvaru $s_1 \wedge s_2$, vznikl jako obraz $\uparrow \{s_1, s_2\} \in \mathscr{U}(P)$. Takže v $\mathscr{U}(P)$ se jednalo o supremum $sup\{\uparrow s_1, \uparrow s_2\}$, což je ale totéž, co sjednocení dvou podmnožin $P$, a sice $\uparrow s_1 \cup \uparrow s_2$. Což objasňuje, proč cestu popisuje sjednocení jednotlivých kritérií.

Naopak spojení dvou termů má význam konjunkce. Bude-li term například ve tvaru spojení dvou literálů $s_1 \vee s_2$, musí být dva pixely spjaty s mírou $s_1$ *a zároveň* $s_2$. Přesně toto říká druhá ekvivalence uvedené věty, tedy že z jednoho pixelu do druhého musí existovat

cesta s termem $s_1$, ale zároveň i cesta s termem $s_2$, ovšem není nutné, aby ty cesty byly stejné.

Dosud uvedené věty byly stále analýzou relace $\Phi$, což znamená, že posuzovaný bod $z: (x, y, z) \in \Phi$ leží v oblasti $\Theta(x, \tau)$ určené body $x$ a $y$. Z poslední věty tedy lze odvodit postup, jak vytvořit oblast $\Theta$. Pro libovolný term tvaru $\tau = \bigvee_{i\in\{1,\cdots,n\}} \bigwedge_{j\in\{1,\cdots,m_i\}} s_{i,j}$ z druhé ekvivalence plyne

$$\Theta(x, \tau) = \bigcap_{j\in\{1,\cdots,m_i\}} \Theta(x, \tau_i),$$

kde $\tau_i = \bigwedge_{j\in\{1,\cdots,m_i\}} s_{i,j}$. Tím se nám zjednodušil problém na hledání $\Theta(x, \tau_i)$ pro každé $i \in \{1, \cdots, n\}$. Bohužel v tomto případě už nelze problém zjednodušit na pouhé sjednocení kritérií. Mějme například term tvaru $s_1 \wedge s_2$. Oblasti $\Theta(x, s_1)$ a $\Theta(x, s_2)$ obsahují pixely, do nichž se lze dostat pomocí jednoho z těchto kriterií, ale vybereme-li bod například z $\Theta(x, s_1)$ a vezmeme v úvahu i literál $s_2$, lze najít další pixely spjaté s oním vybraným bodem s mírou $s_2$, které ale nemusí ležet v $\Theta(x, s_2)$. Tudíž $\Theta(x, \bigwedge_{j\in\{1,\cdots,m_i\}} s_{i,j}) \supseteq \bigcup_{j\in\{1,\cdots,m_i\}} \Theta(x, s_{i,j})$, proto pouhé sjednocení nestačí.

Místo toho je nutné použít iterační algoritmus postupného zvětšování oblasti. Začneme s $T = \{x\}$. V každém kroku vybereme $u \in T$ a najdeme všechny okolní pixely $v \in V$ splňující $\xi(u, v) \geq s_{i,j}$ alespoň pro některé $j \in \{1, \cdots, m_i\}$. Všechny takové pixely $v$ přidáme do $T$. Díky tomu, že pracujeme nad konečným obrazem, algoritmus skončí a výsledná množina pak bude $T = \Theta(x, \tau_i)$.

Teoreticky tento algoritmus zní jednoduše, ale vyvstává problém s hledáním vyhovujících pixelů $v$. Hrozí totiž, že v každém kroku bude nutné projít celý obraz, což by mohlo způsobit značné zhoršení výpočetní náročnosti. Tomuto bude možné předejít, jen pokud zapojíme více znalostí daného problému a jeho vlastností. Konkrétně zde omezíme algoritmus použitím hlubších znalostí o struktuře kritéria $\xi$. Již dříve jsme kritérium specifikovali jako uspořádanou $n$-tici podobností v jednotlivých vlastnostech.

V zájmu zjednodušení algoritmu budeme nadále požadovat, aby první hodnota kritéria popisovala rozdíl vzdálenosti $d$, samozřejmě s převráceným uspořádáním, což je ovšem rozumné pro všechny parametry kritéria, neboť o jakoukoliv vlastnost se jedná, malý rozdíl hodnot značí velkou podobnost a naopak velký rozdíl svědčí o malé podobnosti. Takže budeme uvažovat jen kritéria tvaru $s = (d, a_1, \cdots, a_{k-1})$.

V algoritmu lze toto omezení uplatnit následujícím způsobem. Místo prohledávání celého obrazu stačí pixely $v$ vyhovující podmínce $\xi(u, v) \geq s_{i,j}$ hledat pouze v (relativně k velikosti celého obrazu) malém okolí pixelu $u$. Toto okolí určíme podle vzdálenosti $d_{i,j}$ příslušející literálu $s_{i,j}$ a samozřejmě podle zvolené metriky. Samozřejmě je zbytečné jednou akceptovaný pixel porovnávat s dalšími literály. Dalšími úvahami o seskupování literálů v klauzuli dle parametru vzdálenosti bychom mohli dosáhnout prohledávání pro specifické klauzule jen v daném rozsahu vzdáleností. Tímto by se dalo dosáhnout maximálního zefektivnění, pokud by ovšem sama obsluha nebyla náročnější, než jednodušší vyhledávání.

## 4.2. Přípustné množiny

Naším cílem je v obraze najít objekt $\Theta(x, \tau)$ obsahující výchozí pixel $x$, který je uzavřen vzhledem k podobnosti s mírou alespoň $\kappa(x, y)$. Co se však stane, když vybereme jiný výchozí pixel? Pokud vybereme jiný pixel z dané oblasti $z \in \Theta(x, \tau)$, nutně získáme opět stejnou oblast, vybereme li však bod mimo toto $\Theta$, získáme zcela jinou oblast, která však nemá s tou předchozí žádné společné pixely. Protože $\forall u \in V \setminus \Theta(x, \tau)$ je $\kappa(x, u) < \tau$, jinak by takový bod musel být v dané oblasti. Takže nutně $\forall z \in \Theta(x, \tau)$ je $\kappa(z, u) < \tau$, jinak bychom se během algoritmu budování $\Theta$ dostali až do $u$, takže $\forall z \in \Theta(x, \tau)$: $z \notin \Theta(u, \tau)$

**Věta 4.2.** *Pro daný digitální obraz $(V, \pi, f)$ s kritériem $\xi : V^2 \to P$ splňujícím obě dodatečné vlastnosti pak pro každý term $\tau \in \mathscr{W}(P)$ existuje rozklad $V$, jehož třídy jsou uzavřeny vůči podobnosti alespoň s mírou $\tau$.* *[6, str. 345]*

**Definice 4.3.** *Třídy ekvivalence z věty 4.2 nazveme* přípustné množiny. *Systém všech přípustných množin označíme $\mathscr{A}$.*
*Pro danou množinu $X \subseteq V$ definujeme $\mathscr{A}_X = \{B \in \mathscr{A} \mid X \subseteq B\}$.* *[6, str. 345]*

$\mathscr{A}$ tedy v podstatě představuje systém všech možných fuzzy objektů v obraze pro všechny možné hodnoty prahu. Pak $\mathscr{A}_X$ je soubor všech fuzzy objektů obsahujících množinu $X$.

**Tvrzení 4.1.** *Ke každé neprázdné podmnožině $X \subseteq V$ existuje nejmenší přípustná množina $\Omega(X)$ obsahující $X$. Jestliže $X = \{x, y\}$ pro nějaké pixely $x, y, \in V$, pak $\Omega(X) = \Theta(x, \kappa(x, y))$.* *[6, str. 345]*

Doposud jsme se snažili výběrem jednoho, pro objekt velmi charakteristického pixelu, postihnout celý objekt, zároveň s nastavením největší možné odlišnosti vybráním druhého pixelu, co nejméně podobného prvnímu, ale stále s velkou jistotou náležejícího objektu. Při použití reálných snímacích zařízení se ale ani hladká plocha nemusí zobrazit jednolitě, v obraze se může projevovat například šum či artefakty způsobené datovou kompresí nebo zmenšením obrazu. Proto pouhé dva body nemusí charakterizovat objekt dostatečně přesně. Zavedení přípustných množin nám poskytuje aparát, jak charakterizovat objekt nějakou jeho podmnožinou. Nejjednodušší případ, dvouprvková množina $X$, nás vrací k původnímu přístupu, toto je tak jeho velmi rozumným rozšířením. Jak však postupovat, přidáme-li další bod? Odpověď je v následující větě.

**Věta 4.3.** *Mějme pixely $x, y, z \in V$, pak*

$$\Omega(x, y, z) = \Theta(x, \kappa(x, y) \wedge \kappa(x, z)).$$

*[6, str. 345]*

Podobně můžeme přidávat body v podstatě do libovolného počtu, z čehož vyplývá následující věta.

**Věta 4.4.** *Mějme množinu $X = \{x_0, x_1, \cdots, x_n\} \subseteq V$, pak*

$$\Omega(X) = \Theta(x, \bigwedge_{i=1}^{n} \kappa(x_{i-1}, x_i)).$$

*[6, str. 345]*

Metoda termového prahování je velmi elegantní, neboť velmi komplexně analyzuje podobnost pixelů a rozhodne o podobě hledaného objektu jen na základě několika výchozích bodů. Zjistili jsme, že hledaný objekt tvoří právě jedna z přípustných množin a představili algoritmus pro jeho nalezení. Celá metoda však závisí na znalosti spjatostního termu.

## 4.3. Určení spjatostního termu

Pracujeme nad digitálním obrazem $(V, \pi, f)$ s kriteriem $\xi : V^2 \to P$. Máme vybránu množinu bodů, o které jsme přesvědčeni, že náleží hledanému objektu. Naším cílem je nalézt nejmenší přípustnou nadmnožinu, jež bude hledaným objektem. Jakmile známe požadované spjatostní termy, nejedná se o náročný úkol. Zde si uvedeme algoritmus pro jejich nalezení, ale jak brzy zjistíme, tato úloha je o poznání složitější.

Začněme výběrem dvou pixelů $x, y \in V$ reprezentujících hledaný objekt. Chceme nalézt jejich spjatostní term ve tvaru $\kappa(x, y) = \bigvee_{\langle x=x_0,x_1,\cdots,x_n=y\rangle} \bigwedge_{1\leq i\leq n} \xi(x_{i-1}, x_i)$.

Spojení značí, že bychom měli hledat přes všechny cesty z $x$ do $y$, což však vede na algoritmus principiálně jednoduchý, ale s neúnosně velkou složitostí, která dosahuje až řádu $\mathcal{O}(3^{|V|})$, a to jen při zvolení jednoduché 4přilehlosti.

Místo prohledávání všech cest, které nelze nijak rozumně omezit zároveň s požadavkem, že výsledkem bude největší možný term, se pokusíme posoudit všechny termy ve $\mathscr{W}$. To se může zdát jako neřešitelný úkol, neboť svaz $\mathscr{W}$ ani explicitně neznáme a ani se nebudeme snažit jej generovat. Vzhledem k rozsáhlosti množiny $P$, která může mít až $\frac{|V|^2}{2}$ prvků, by to totiž nebyl efektivní postup, v podstatě by to znamenalo ověřit každou podmnožinu $P$. Stačí však vědět, že tento svaz existuje a můžeme využít jeho vlastností.

Víme, že obsahuje největší prvek a známe i jeho podobu $\top = \xi(x, x)$, která v praxi bude vypadat jako $k$-tice samých nul. Algortimus proto začneme s termem $\tau = \top$ a postupně jej budeme snižovat, čímž se bude zvětšovat příslušná oblast $\Theta$.

Víme už, že hledaný term $\kappa(x, y) = \bigvee_{\langle x=x_0,x_1,\cdots,x_n=y\rangle} \tau_i$ je spojením jednotlivých klauzulí a výsledná hledaná oblast $\Theta(x, \kappa(x, y)) = \bigcap_{\langle x=x_0,x_1,\cdots,x_n=y\rangle} \Theta(x, \tau_i)$ je průnikem oblastí vzniklých z těchto klauzulí. Proto stačí hledat pouze termy tvaru konjunktivních klauzulí. Je nutné však najít všechny, které jsou dostatečně malé, aby příslušná oblast s bodem $x$ obsahovala i bod $y$, takové termy nazveme *cílové*. Je tedy postačující, když se budeme pohybovat pouze po konjunkcích literálů.

Dále si uvědomme, že není nutné procházet všechny menší termy, ale jen ty, které povedou ke zvětšení oblasti. Term lze snížit jeho konjunkcí s nějakým literálem, neboť pro jakékoliv termy $\varphi, psi$ vždy platí $\varphi \wedge \psi \leq \varphi$ a $\varphi \wedge \psi \leq \psi$. Vybereme tedy všechny literály, které povedou ke zvětšení oblasti. Pochopitelně se musí jednat o literály, které současný term neobsahuje. Pokud by totiž bylo možné přidat do oblasti $\Theta$ nějaký bod přidáním literálu $l$, který už však je v $\tau$, museli bychom se do tohoto bodu dostat už při budování $\Theta$. Zároveň se ale budeme snažit zvětšovat oblast co nejméně, to znamená, že ze zvětšujících literálů vybereme jen ty maximální, tuto množinu označíme $M(\tau)$. Procházení všech literálů způsobí větvení algoritmu.

Je zřejmé, že platí $\tau_1 \leq \tau_2 \Rightarrow \Theta(x, \tau_1) \supseteq \Theta(x, \tau_2)$. Proto jakmile zjistíme, že některý term dal vzniknout oblasti zahrnující bod $y$, není třeba kontrolovat všechny menší, protože příslušné oblasti také obsahují $y$.

Shrneme-li tyto úvahy, máme algoritmus postupného zvětšování původně jednoprvkové oblasti $\{x\}$ maximálními možnými termy. Prohledávání každé z větví stavového prostoru lze ukončit, pokud dosáhneme bodu $y$ nebo pokud získáme term menší, než některý cílový.

### Algoritmus hledání spjatostního termu

Ve výchozím stavu máme frontu $F$ cílových termů, která je prázdná, a s frontu $Q$ posuzovaných termů, která obsahuje pouze term $\top$.

1. Odebereme term $\tau$ ze začátku fronty $Q$
2. Najdeme množinu $M(\tau)$, kterou tvoří maximální prvky množiny $\{\xi(u, v) \mid u \in \Theta(x, \tau),\ v \in V \setminus \Theta(x, \tau)\}$
3. $\forall l \in M(\tau)$ zkontrolujeme, zda je v $F$ nějaký větší term než $\tau \wedge l$.
   Pokud ne, zkontrolujeme, zda $y \in \Theta(x, \tau \wedge l)$:
   - pokud ano, $\tau \wedge l$ vložíme do $F$ a z $F$ odstraníme všechny menší termy,
   - pokud ne, $\tau \wedge l$ vložíme do $Q$, pokud už tam není.
4. Je-li $Q$ prázdná, algoritmus končí. V opačném případě pokračujeme opět bodem 1.

Na konci $F$ obsahuje jen maximální cílové termy, což jsme zajistili kontrolou větších termů v úvodu třetího kroku. Pokud bychom v $F$ nějaký větší term našli, znamenalo by to, že i právě analyzovaný term je schopný dosáhnout $y$, proto není třeba jej kontrolovat, ale ani zapisovat do $F$, protože menší term neovlivní konečné spojení.

Díky procházení množiny $M(\tau)$ procházíme obraz jen pomocí literálů, které jsou nutné k tomu, abychom zvětšili oblast $\Theta$, bez čehož logicky vůbec nelze dosáhnout bodu $y$. Navíc tím eliminujeme termy, které nevedou k cíli, nebo které lze „obejít“ tak, že je stejně získáme v pozdějších fázích algoritmu. Proto tvrdíme, že všechny cílové termy jsou buď v seznamu $F$ nebo jsou menší, což vede k tvrzení následujícímu.

**Tvrzení 4.2.** *Po skončení algoritmu je* $\kappa(x, y) = \bigvee F$. *[6, str. 346]*

Na tomto tvrzení stojí celá aplikovatelnost algoritmu. Důkaz staví na tom, že ukážeme, že že právě cesty s maximálním termem dostanou svůj term do $F$. Je však poněkud zdlouhavý, ale lze jej snadno dohledat v [6, str. 346].

Algoritmus je bohužel velmi pomalý, proto si později představíme i několik možností jeho zrychlení. V nehorším případě je nutné počítat s každou možnou přípustnou množinou, neboť jí přísluší některý term $\tau$, který je třeba analyzovat. Množinu $\mathscr{A}$ tvoří nejvýše všechny podmnožiny $V$, což je počet $2^{|V|}$. Největší počet termů, které lze zapsat do seznamu $Q$, je tedy také $2^{|V|}$, ale složitost každého kroku algoritmu je řádově nižší, proto celková složitost bude nejhůře řádu $\mathcal{O}(2^{|V|})$. Což je sice asymptoticky lepší, než $\mathcal{O}(3^{|V|})$, ale stále je to složitost velmi špatná.

Nejhorší případ však také znamená, že, vzhledem ke konstrukci množiny $M$, se přidáním literálu k termu příslušná oblast zvětší o jediný bod. Většinou však budeme pracovat s obrazy, které obsahují spíše hladké oblasti, než absolutně náhodný šum. Proto oblasti porostou rychleji a přípustných termů bude ve skutečnosti nižší počet. Dalším předpokladem nejhoršího možného scénáře je, že pro dosažení cílového bodu je nutné projít všechny termy. Takový scénář však nastane, jen když budou cílové termy jen ty minimální, například pokud by všechny termy byly nesrovnatelné. ani takový případ však ve skutečném obraze nejspíše nenastane. Skutečné výpočetní časy tak zdaleka nedosahují horní hranice.

Tímto jsme položili kompletní teoretický základ metody vícekriteriální segmentace. Nyní tedy lze přistoupit k její prezentaci a možným vylepšením

# 5. Implementace

V zájmu lepší ilustrace dosud představené teorie byla provedena implementace algoritmu pro hledání spjatostních termů. Výstupem je přiložený program, jenž byl naprogramován v jazyce C++ s uživatelským rozhraním ve frameworku Qt.

Samotný kód se skládá z několika částí. Soubor `main.qml` má na starost uživatelské rozhraní programu, které zde však nebudeme rozebírat. V souboru `FuzzyAlgorithms.h` se nachází obsluha všech metod, zatímco samotné algoritmy se nacházejí v souboru `FuzzyAlgorithms.cpp`. Propojení uživatelského rozhraní a vnitřní logiky programu se děje v souboru `main.cpp`. Další hlavičkové a zdrojové soubory obsahují některé pomocné struktury. Těmi jsou `obrazek`, `pixel`, `term` a `indicator`.

## 5.1. Segmentace založená na afinitě

V programu je taktéž implementován algoritmus afinitní segmentace s afinitou

$$\psi(x,y) = \left( \frac{2}{1 + |\frac{f(x)}{255} - \frac{f(y)}{255}|} - 1 \right),$$

kde $f(x)$ a $f(y)$ jsou hodnoty jasu obou pixelů v rozsahu $\langle 0, 255 \rangle$. Je zde možnost vybrat výchozí bod a zvolit práh, ale vzhledem ke zvolenému kriteriu je metoda poměrně necitlivá a práh je nutné volit velmi vysoký. Případně lze vybrat dva body v obraze a práh se zvolí automaticky. Vstup zpracovává metoda `connMap()`, kde se vytvoří instance obrazu. Ten je procházen následujícím algoritmem dle [2, str. 75].

**Dynamický program pro fuzzy segmentaci.** Máme frontu pixelů $Q$, nový obraz $(V, \pi, m_x)$ a výchozí bod $x$. Ve výchozím stavu obsahuje $Q$ pouze bod $x$ a hodnoty spjatostní mapy jsou $m_x(x) = 1$ a $m_x(u) = 0,\ \forall u \in V, u \neq x$.

1. Vyjmeme $u$ z $Q$.
2. $\forall v \in V$ takové, že $(u, v) \in \pi$ a provedeme následující:
   - nastavíme $s = \min\{m_x(u),\ \psi(u, v)\}$,
   - pokud $s > m_x(v)$, pak $v$ vložíme do $Q$ a nastavíme $m_x(v) = s$.
3. Je-li $Q$ prázdná, algoritmus končí. V opačném případě pokračujeme opět bodem 1.

Všechny výstupy programu se ukládají do složky `output`, jež je umístěna ve složce s programem. Název souboru se skládá z času spuštění dané metody a některých parametrů. Zde se taktéž uloží výsledná spjatostní mapa $(V, \pi, m_x)$.

## 5.2. Vícekriteriální segmentace

Pro účely ilustrace metody je implementována možnost hledání fuzzy oblasti pouze pro dva či tři výchozí body. Zahrnuli jsme zde však možnost výběru metriky. Zpracované metriky jsou součtová, též zvaná manhattanská

$$d(x, y) = |x_1 - y_1| + |x_2 - y_2|,$$

maximová metrika

$$d(x, y) = \max\{|x_1 - y_1|, |x_2 - y_2|\},$$

a aproximace euklidovské metriky

$$d(x, y) = \text{round}\left(\sqrt{|x_1 - y_1|^2 + |x_2 - y_2|^2}\right).$$

Dále lze nastavit segmentaci pomocí dvou nebo tří kriterií, která lze zvolit jako rozdíl hodnot jasu nebo rozdíl hodnot v jednotlivých barevných kanálech obrazu ve standardu RGB. Pro přepočet barevných hodnot (označených $r$, $g$ a $b$) na jas (označen $f$) používáme vztah

$$f(x) = \text{round}\left(0,2126 \cdot r(x) + 0,7152 \cdot g(x) + 0,0722 \cdot b(x)\right).$$

Obsluha této segmentační metody se provádí v metodě `connTerm()`. Zde dochází k volání výpočetního algoritmu, který se nalézá v metodě `connectednessTerm()`. Dále v případě, že je zvolena segmentace podle více než dvou bodů, se zde provádí výběr bodů pro jednotlivé kroky výpočtu termu a poté je zpracován výsledný term z těchto částečných. Nakonec je volána metoda `BuildTheta()`, která k vzniklému termu přiřadí oblast $\Theta(x, \tau)$.

Hledání spjatostního termu probíhá přesně podle algoritmu uvedeného v předchozí kapitole. Výpočet je však často velmi zdlouhavý. Přestože se nejspíše ani zdaleka nepřiblížíme k teoretickému hornímu limitu, vysoký počet iterací způsobí téměř neúnosnou délku celého procesu. Naším hlavním problémem je tedy výpočetní náročnost. Požadavky na paměť jsou totiž řádově nižší. I proto všechny výsledky ilustrujeme na poměrně malých obrazech.

V každé iteraci je několikrát vytvořena oblast $\Theta$ pro termy $\tau$ a několik termů $\tau \wedge l$. Ty jsou ale koncipovány jako seznamy souřadnic ve formátu dvojice integerů. Rozměr těchto seznamů je tedy nejvýše $|V| \cdot 8B$. Běžně komerčně dostupné fotoaparáty mají snímací čipy s rozlišením do $50Mpx$, kde může vzniknou oblast $\Theta$, která v paměti zabere místo o velikosti nejvýše necelých $400MB$. To je při současných velikostech počítačových pamětí stále únosná hodnota, avšak výpočetní náročnost u tak velkého obrazu by byla nesrovnatelně vyšší. Proto zde představíme i některé možnosti, jak výpočet zrychlit.

### 5.2.1. Výstavba množiny $\Theta$ z konjunktivní klauzule.

Algoritmus je naprogramován v metodě `GrowTheta()`. Výstavba probíhá iterativně z množiny obsahující výchozí bod $T = \{x\}$

1. Projdeme všechny body v seznamu $T$, prvky indexujeme od $i = 0$.
2. Zjistíme okolí $i$-tého bodu $O(T_i)$.

3. $\forall u \in O(T_i) \setminus T$ zjistíme, zda $\xi(u, T_i) \geq \tau$
   pokud ano, vložíme $u$ na konec $T$.

4. Zvýšíme index o 1.
   Pokud je $i < |T|$, pokračujeme bodem 1, jinak algoritmus končí a $\Theta(x, \tau) = T$.

Algoritmus pokračuje, dokud se oblast zvětšuje. Navíc tím, že index stále roste, přestože se zvětšuje velikost seznamu $T$, nedochází ke zbytečné kontrole bodů, jejichž okolí jsme již jednou prohledávali. Velikost okolí jsme nastavili konstantní pro celý průběh, a to podle maximální vzdálenosti ze všech literálů v dané klauzuli. Nabízí se sice optimalizace, kde použijeme proměnlivá okolí, či skupiny literálů budeme porovnávat na mezikruží, ale porovnání hodnot ve všech bodech je asi srovnatelně náročné, jako by byl aparát, který by taková okolí generoval.

Jisté zefektivnění však stále lze provést. V kroku 3 algoritmu hledání spjatostního termu generujeme oblast $\Theta(x, \tau \wedge l)$, která musí být nadmnožinou oblasti $\Theta(x, \tau)$ vygenerované na začátku iterace, neboť vždy platí $\tau \wedge l \leq \tau$.

Uvědomme si nyní, že všechny body $\Theta(x, \tau)$ bychom museli vlastně vygenerovat podruhé. Proto nebudeme oblast $\Theta(x, \tau \wedge l)$ generovat z jednoprvkové množiny $T = \{x\}$, ale jako výchozí stav algoritmu zde použijeme $T = \Theta(x, \tau)$. Stále dojde ke kontrole všech bodů $T$, zda v jejich okolí neleží bod, jehož spjatost s kontrolovaným $T_i$ je lepší než $\tau \wedge l$, ale nebudou nadále zbytečně porovnávány s body původní oblasti $\Theta(x, \tau)$.

V současné podobě implementace je $\Theta(x, \tau)$ uložen v paměti až do přepsání v další iteraci. Proto taková úprava je vzhledem k paměťové náročnosti málo významná.

### 5.2.2. Vytvoření množiny $M(\tau)$

Po vytvoření množiny $\Theta(x, \tau)$ následuje krok hledání všech literálů, které „spojí vnitřek oblasti z vnějškem“. Tedy vytvoříme množinu $\{\xi(u, v) \mid u \in \Theta(x, \tau),\ v \in V \setminus \Theta(x, \tau)\}$, ze které pak vybereme všechny maximální prvky, čímž vznikne množina $M(\tau)$. Toto je v základním tvaru jeden z nejnáročnějších kroků každé iterace, jelikož je nutné projít celý obraz.

Pracujeme se seznamem $S$ všech zjištěných kritérií, který je ve výchozím stavu prázdný.

1. Zkontrolujeme každý bod $T_i$ oblasti $\Theta(x, \tau)$.

2. Projdeme každý bod $v_j \in V \setminus \Theta(x, \tau)$

3. Zjistíme hodnotu kriteria $\xi(T_i, v_j)$. Tuto hodnotu uložíme do seznamu $S$.

Počet operací zde může dosahovat až $\frac{|V|^2}{4}$. Stejné hodnoty může dosahovat i velikost seznamu $S$, Který bychom následně museli uspořádat. Proto bude jednodušší udržovat tento seznam malý a takový, aby obsahoval jen maximální hodnoty. Začneme tedy hodnotu kritéria před zapsáním porovnávat s obsahem seznamu. Nahradíme bod 3 následujícími kroky.

3'. • Zkontrolujeme, zda je v $S$ nějaký větší literál, než $\xi(T_i, v_j)$.
• Pokud není, uložíme $\xi(T_i, v_j)$ do $S$ a všechny menší a stejné literály odstraníme.

Po skončení algoritmu je $M(\tau) = S$.

Seznam $S$ tady udržujeme malý, neboť možný počet neporovnatelných kritérií je podstatně menší, než $|V|^2$. Proto není ani příliš náročná jeho průběžná kontrola. Nyní zde představíme dva možné přístupy ke zrychlení tohoto kroku. Oba následující přístupy jsou naprogramovány v metodě `getMaxM()`.

### Omezení prohledávané vzdálenosti

V programu je zahrnuta možnost omezení maximální vzdálenosti od kontrolovaného bodu $T_i$ při prohledávání zbylé části obrazu $V \setminus \Theta(x, \tau)$. Po zvolení této možnosti lze v obraze vybrat dva body, jejichž vzdálenost při dané metrice je použita jako maximální vzdálenost pro prohledávání.

Vycházíme zde z úvahy, že mezi všemi zadanými body se uvnitř souvislého objektu dá přejít i $\pi$-spojitou cestou nebo cestou s jen malými skoky po nepřilehlých pixelech, proto nejsou nutné velké vzdálenosti v jednotlivých literálech. Naopak pokud je objekt nesouvislý, lze určit, jak vzdálené části obrazu ještě budeme považovat za součást objektu. Taková nesouvislost může vzniknout například překrytím jiným objektem.

Vždy je totiž možné, že obraz obsahuje například oblast podobné barvy jako hledaný objekt, která je ale od objektu oddělena nějakým skokem. Zde můžeme předpokládat, že je-li jejich vzdálenost moc velká, jedná se o součást pozadí, nikoliv o objekt. Přesto se však jedná pouze o *heuristiku*, ne o exaktní metodu.

Vynecháním některých klauzulí se výsledný term zmenší, což vede ke zvětšení oblasti. Jsme však schopni odhadnout dolní i horní hranici výsledné oblasti. Podobným tématem se zabývá [6, str. 348].

Výsledný term ve tvaru $\kappa(x, y) = \bigvee_{i \in I} \tau_i(x, y)$ je spojením konjunktivních klauzulí $\tau_i(x, y)$. Omezení prohledávané vzdálenosti lze formalizovat jako vynechání všech termů $\tau_i(x, y) \leq \lambda$, kde term $\lambda$ nastavíme jako jediný literál $\lambda = (d_{max}, 0, \cdots, 0)$. Zde vidíme, že každý literál se vzdáleností větší nebo stejnou jako $d_{max}$ bude menší než $\lambda$, nezávisle na ostatních hodnotách. Ostatní literály s menší hodnotou vzdálenosti budou buď menší nebo nesrovnatelné.

Klauzule rozdělíme na dvě skupiny, ty menší než $\lambda$, které budou indexovány $\widehat{I} = \{i \in I \mid \tau_i(x, y) \leq \lambda\}$, a větší než $\lambda$ spolu s nesrovnatelnými, indexovány $\widetilde{I} = I \setminus \widehat{I}$.

Nyní označíme $\widetilde{\kappa}(x, y) = \bigvee_{i \in \widetilde{I}} \tau_i(x, y)$ a $\widehat{\kappa}(x, y) = \bigvee_{i \in \widehat{I}} \tau_i(x, y)$. Následující vztahy jsou zřejmé.

$$\begin{aligned}
&\widehat{\kappa}(x, y) \leq \lambda \\
&\widehat{\kappa}(x, y) \leq \kappa(x, y) \quad \text{a} \quad \widetilde{\kappa}(x, y) \leq \kappa(x, y) \\
&\widehat{\kappa}(x, y) \vee \widetilde{\kappa}(x, y) = \kappa(x, y)
\end{aligned}$$

Z těchto vztahů také vyplývá

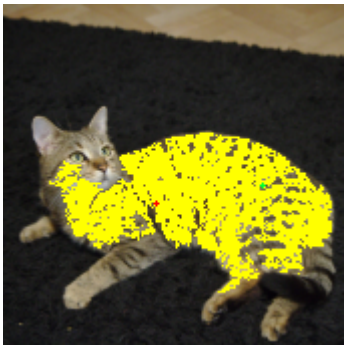
$$\widetilde{\kappa}(x,y) \leq \kappa(x,y) \leq \widetilde{\kappa}(x,y) \vee \lambda.$$

[6, str. 348]

Proto také platí

$$\Theta\left(x, \widetilde{\kappa}(x,y)\right) \supseteq \Theta\left(x, \kappa(x,y)\right) \supseteq \Theta\left(x, \widetilde{\kappa}(x,y) \vee \lambda\right).$$

Jsme tedy schopni ohraničit velikost hledané oblasti, ale nejsme schopni určit přesné řešení.

1)  2) 

Obrázek 5.1: *1) segmentace bez použití a 2) s použitím omezení prohledávané vzdálenosti.*

Na obraze 5.1 jsme provedli segmentaci bez omezení (1) a s použitím omezení prohledávané vzdálenosti (2). Objekt vzniklý segmentací bez omezení je znatelně menší.
Výsledný segmentační term pro 1) je $\kappa(x,y) = [(8,0) \wedge (1,3)] \vee [(8,0) \wedge (2,2)] \vee [(8,0) \wedge (4,1)] \vee [(7,0) \wedge (1,4)] \vee [(13,0) \wedge (2,1)] \vee (1,11) \vee [(17,0) \wedge (1,1)] \vee [(4,1) \wedge (3,3)] \vee (7,1) \vee [(6,1) \wedge (1,4)] \vee (2,7) \vee [(3,3) \wedge (7,0)] \vee [(1,3) \wedge (5,2)] \vee [(1,7) \wedge (4,2)] \vee [(3,3) \wedge (1,10)] \vee (3,6)$, který je však poměrně rozsáhlý.

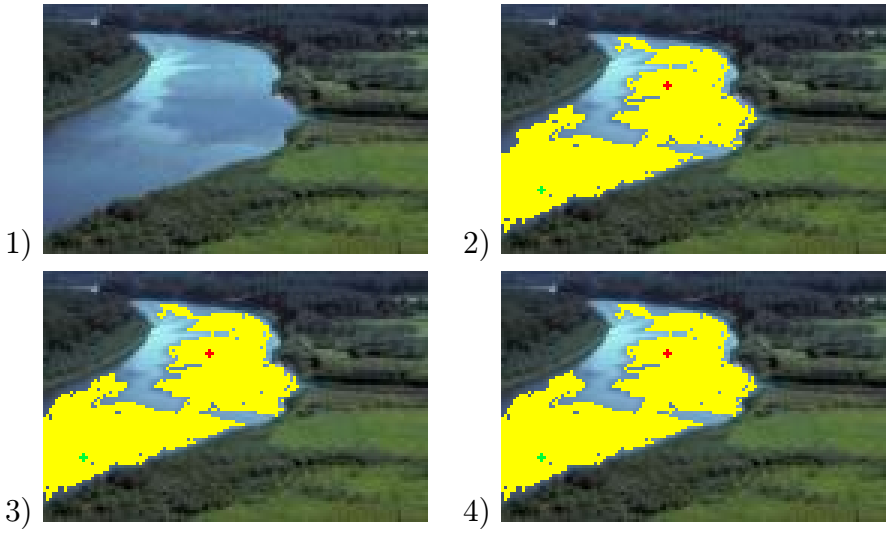
Po omezení vzdálenosti na $d_{max} = 6px$ ve 2) jsme získali omezený term $\widetilde{\kappa}(x,y) = (1,11) \vee [(4,1) \wedge (3,3)] \vee [(6,1) \wedge (1,4)] \vee (2,7) \vee [(1,3) \wedge (5,2)] \vee [(1,7) \wedge (4,2)] \vee [(3,3) \wedge (1,10)] \vee (3,6)$. Ten je zcela zřejmě očištěn o všechny klauzule obsahující nějaký parametr vzdálenosti větší než 6. Výsledný objekt je větší, ale překvapivě lépe vystihuje svou skutečnou předlohu. To nás může vést k domněnce, že tato metoda může být v některých případech výhodná, neboť jsme zredukovali term i čas výpočtu a zároveň jsme velmi dobře postihli hledanou oblast.

Obrázek 5.2 zachycuje segmentaci obrazu o velikosti $128 \times 84$. Na něm si předvedeme časovou efektivitu metody.

Bez omezení jsme ve 2) získali spjatostní term $\kappa(x,y) = [(1,4)] \vee [(1,2) \wedge (6,0)] \vee [(5,0) \wedge (1,3)] \vee [(4,1)] \vee [(3,0) \wedge (1,3) \wedge (2,2)] \vee [(2,2) \wedge (3,1)] \vee [(2,3)] \vee [(2,1) \wedge (9,0)] \vee [(1,1) \wedge (10,0)] \vee [(5,0) \wedge (2,2)]$. Celková doba zpracování byla 16 min 24 s zároveň s celkovým průběhem 78 iterací algoritmu.

Ve 3) bylo použito omezení vzdálenosti $d_{max} = 7px$. Získali jsme přibližný term $\widetilde{\kappa}(x,y) = [(1,4)] \vee [(1,2) \wedge (6,0)] \vee [(5,0) \wedge (1,3)] \vee [(4,1)] \vee [(3,0) \wedge (1,3) \wedge (2,2)] \vee [(2,2) \wedge (3,1)] \vee [(2,3)] \vee [(5,0) \wedge (2,2)]$během 66 iterací v celkovém čase 9s.

Pro 4) bylo použito omezení vzdálenosti $d_{max} = 4px$. Přibližný term $\widetilde{\kappa}(x,y) = [(1,4)] \vee [(4,1)] \vee [(3,0) \wedge (1,3) \wedge (2,2)] \vee [(2,2) \wedge (3,1)] \vee [(2,3)]$byl vypočten v 50 iteracích za dobu pouhých 3s.




Obrázek 5.2: *1) fotografie řeky Innoko, [Jo Keller, public domain], 2) segmentace bez použití omezení, 3) s omezením vzdálenosti $d_{max} = 7px$, 4) s omezením vzdálenosti $d_{max} = 4px$.*

Vzniklé oblasti jsou na první pohled stejné, vyskytly se pouze nepatrné rozdíly, ale došlo k významnému zkrácení doby výpočtu. Zatím se tedy tento přístup zdá jako výhodný.

### Prořezávaní prohledávaného prostoru

Chceme-li zrychlit generování množiny $M(\tau)$, ale bez toho, abychom ztratili přesné řešení, je nutné přijít se sofistikovanější metodou omezování prohledávané oblasti.

Představme si, že porovnáváme vzdálenost a rozdíl jasu. Je-li oblast $\Theta$ například uprostřed obrazu, vybereme první pixel v $\Theta$ a postupně procházíme celý obraz. V krajích obrazu nejspíše budou pixely, jenž budou vracet velice špatné hodnoty kritéria, a čím více se budeme blížit ke středu obrazu, tím lepší hodnoty budeme dostávat, až v nejbližším okolí $\Theta$ získáme hodnoty, které budou tvořit maximum seznamu, a tedy $M(\tau)$. Zdá se zbytečné projít druhou polovinu obrazu, když je téměř vyloučené, že bychom dostali jakékoliv lepší hodnoty. Je však nějaká šance, abychom si mohli být naprosto jisti a nevynechali tak nějaký důležitý literál?

Zde však můžeme přistoupit k jistému třídění na základě dosud získaných hodnot. Zkrátka přidáme-li do seznamu $S$ term tvaru $\lambda = (d, 0, \cdots, 0)$, je již naprosto vyloučeno, aby se vyskytoval nějaký větší či nesrovnatelný term ve vzdálenosti větší než $d$. Průběžně při průchodu algoritmu tedy budeme ořezávat velikost prohledávaného okolí. Modifikujeme tedy kroky 2 a 3', čímž dostaneme upravený algoritmus.

Pracujeme se seznamem $S$ všech zjištěných kritérií, který je ve výchozím stavu prázdný, na začátku je $d_{max}$ rovno maximální vzdálenosti v obraze.

1. Zkontrolujeme každý bod $T_i$ oblasti $\Theta(x, \tau)$.

2'. Projdeme každý bod $v_j \in O_{d_{max}}(T_i) \setminus \Theta(x, \tau)$ z okolí $T_i$

3". • Zkontrolujeme, zda je v $S$ nějaký větší literál, než $\xi(T_i, v_j)$.

- Pokud není, uložíme $\xi(T_i, v_j)$ do $S$ a všechny menší a stejné literály odstraníme, navíc, je-li $\xi(T_i, v_j)$ tvaru $(d, 0, \cdots, 0)$ a $d < d_{max}$, obnovíme hodnotu $d_{max} = d$

**Tvrzení 5.1.** *Po skončení algoritmu je* $M(\tau) = S$.

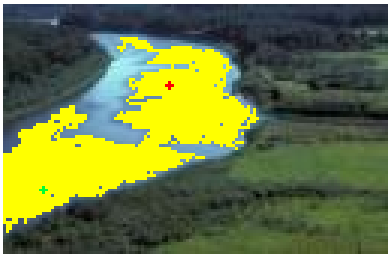
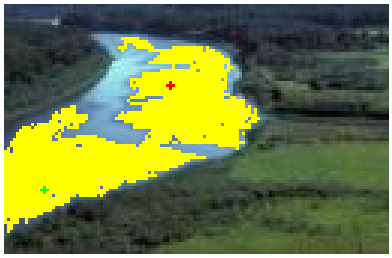
*Důkaz.* Skutečně, existuje-li literál $\xi = (d, 0, \cdots, 0)$, pak pro každý literál tvaru $\zeta = (\Delta, a_1, \cdots, a_{k-1})$, kde $\Delta \geq d$, platí $\zeta \leq \xi$, neboť vždy platí $a_1 \geq 0, \cdots, a_{k-1} \geq 0$. Pamatujme, že uspořádání kritérií je převrácené vůči uspořádání jednotlivých složek.

Hodnoty $a_1, \cdots, a_{k-1}$ mohou být libovolné. Pak jediné další možné tvary literálů mohou být $\zeta = (\delta, a_1, \cdots, a_{k-1})$, kde $\delta < d$. Konkrétně existují literály tvaru $\zeta_1 = (\delta, 0, \cdots, 0)$, pro které platí $\zeta_1 \geq \xi$, a tvaru $\zeta_2 = (\delta, a_1, \cdots, a_{k-1})$, $a_i$ libovolné pro $1 \leq i \leq k$, které jsou s $\xi$ nesrovnatelné.

Dále jakmile $\xi$ vložíme do $S$, bude možné do $S$ přidat už jen literály větší a nebo nesrovnatelné. Víme již, že všechny větší a nesrovnatelné literály jsou tvaru $(\delta, a_1, \cdots, a_{k-1})$, $\delta < d$. Vidíme tedy, že po přidání takového literálu již není možné vložit do $S$ žádný literál se vzdáleností větší nebo rovnou $d$. □

Proto je zbytečné nadále prohledávat ve větší vzdálenosti. Poloměr okolí tak lze zmenšovat za běhu algoritmu.

Oba tyto přístupy jsou koncipovány jako volitelné, lze si vybrat metodu zrychlení a porovnat tak jejich účinnost.

1)  2) 

Obrázek 5.3: *1) Segmentace s použitím prořezávání prohledávaného prostoru, 2) segmentace s ořezáváním a zároveň s omezením prohledávané vzdálenosti* $d_{max} = 7px$.

Na obraze 5.3 jsme použili metodu prořezávání prohledávané oblasti. V 1) jsme získali stejný term $\kappa(x, y)$ jako v předchozím případě na obrázku 5.2 bez použití heuristiky. Celkový počet iterací algoritmu se též nezměnil, avšak čas výpočtu se značně snížil, a to na 17s.

Ve 2) jsme zároveň použili heuristiku s omezením $d_{max} = 7px$. Výsledný term $\widetilde{\kappa}(x, y)$ je stejný jako jsme získali v předchozí metodě se stejným omezením, i počet proběhlých iterací zůstal zachován. Snížil se však výpočetní čas na 7s.

Vyvozujeme zde proto závěr, že metoda prořezávání je velmi účinná pro dvouprvková kriteria. Pro víceprvkové kriterium může být výhodnější použít zároveň s heuristikou, neboť se snižuje šance na nalezení literálu s hodnotou $(d, 0, \cdots, 0)$.

## 5.3. Zakázané oblasti

Lidské oko je nejlepším dostupným segmentačním nástrojem, naší snahou je však co největší automatizace segmentačního procesu. Člověk je schopen odlišit i objekt, který je

téměř nerozlišitelně oddělený od svého pozadí. Snadno se může stát, že segmentovaná oblast „přeteče“ do pozadí, či pohltí nějaký další podobný objekt. Člověk však dokáže snadno rozpoznat i objekty, které ani představená metoda nedokáže rozlišit. Lidský mozek si je totiž schopný domýšlet, že nějaká část obrazu nemůže být součástí objektu, na který se dívá. My se takové chování pokusíme částečně napodobit i v následujícím rozšíření uvedené metody.

Dosud jsme vybírali body, o kterých si myslíme, že náleží hledanému objektu, a to navíc tak, aby si byly vzájemně co nejméně podobné. Právě tento výběr simuluje úlohu mozku v analýze informací, které člověk získal svým zrakem, neboť do metody dodává některé faktické znalosti o problému. Zahrnutím dalších informací by se tedy měl model zdokonalit.

V některých situacích se nám může zdát segmentovaný obraz moc velký. Jsme totiž schopni se ze zkušenosti rozhodnout, že výsledek segmentace pokryl i jiný než hledaný objekt. Takovou dodatečnou informaci se pokusíme do metody dodat.

Jsme-li si jisti, že některá část výsledného objektu k němu určitě nepatří, můžeme vybrat jeden z těchto nepatřičných bodů a pokusit se jej ze současného objetu odebrat.

Mějme tedy obraz $(V, \pi, f)$ ve kterém jsme vybrali dva body $x_1$ a $x_2$. Úvaha pro libovolný počet bodů $n$ by byla obdobná, dvojice bodů je však jednodušší pro představu i zápis. Segmentací vznikne fuzzy objekt $\Theta(x_1, \kappa(x_1, x_2))$, který je zároveň přípustnou oblastí. Zjistili jsme však, že tato oblast obsahuje bod $z$, o kterém jsme přesvědčeni, že by k danému objektu příslušet neměl. Pak platí

$$\kappa(x_1, x_2) \leq \kappa(x_1, z) \quad \text{a} \quad \kappa(x_1, x_2) \leq \kappa(x_2, z).$$

Je zřejmé, že takovou oblast zcela jistě nelze popsat jediným termem. Proto nelze vytvořit přípustnou oblast, která obsahuje body $x_1$ a $x_2$, ale nikoliv $z$. K jeho odebrání proto musíme přistoupit jinak.

Cílem je vytvořit oblast jednoznačně určenou zadanými i zakázanými body. Zde se nabízí použít dvojici přípustných oblastí a vytvořit jejich *rozdíl*. První oblastí je původní oblast $\Theta(x_1, \kappa(x_1, x_2))$. Nyní chceme vytvořit druhou oblast obsahující $z$, která ale neobsahuje žádný z bodů $x_1$ a $x_2$. Označme tuto hledanou oblast $\Theta(z, \tau)$. Pak musí platit

$$\tau > \kappa(x_1, z) \geq \kappa(x_1, x_2) \quad \text{a} \quad \tau > \kappa(x_2, z) \geq \kappa(x_1, x_2).$$

Chceme tento term určit jednoznačně a zároveň tak, aby byl co nejmenší, což způsobí, že odečítaná oblast bude největší možná. Předpokládejme, že je tvaru $\tau = \bigvee_{j \in J} \tau_i$. Postačující podmínkou, aby $\tau > \kappa(x_1, z)$ a $\tau > \kappa(x_2, z)$, je

$$\tau_i > \kappa(x_1, z) \quad \text{a} \quad \tau_i > \kappa(x_2, z), \ \forall j \in J.$$

Zároveň ale požadujeme, aby klauzule $\tau_i$ byly co nejmenší.

Pro získání těchto klauzulí můžeme modifikovat algoritmus hledání spjatostního termu. V kroku 3 testujeme, zda oblast $\Theta(x_1, \tau \wedge l)$ obsahuje cílový bod $x_2$ a pokud ano, uložíme term $\tau \wedge l$ mezi cílové termy. Navrhujeme modifikaci, kde kontrolujeme, zda $\Theta(z, \tau \wedge l)$ obsahuje některý z bodů $x_i$ a pokud ano, uložíme $\tau$, jako jednu z hledaných klauzulí.

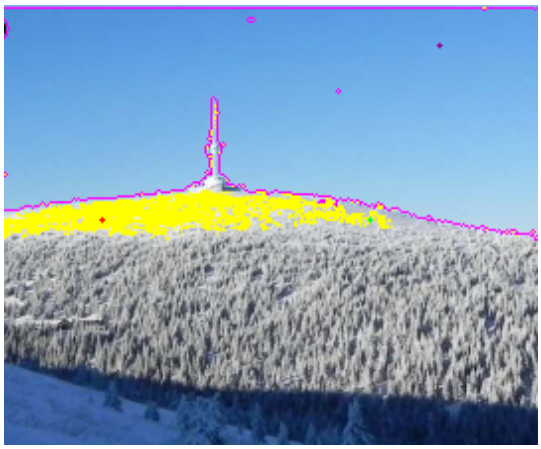
Teoreticky nás sice zajímá každý literál $\lambda$ vetší než $l$, protože klauzule $\tau \wedge \lambda$ by stále tvořila oblast, která nedosahuje žádného z bodů $x_i$, $i \in I$, ale takový literál by nevedl

ke zvětšení oblasti. Je-li totiž $\lambda > l,\ l \in M(\tau)$, pak $\Theta(z,\tau) = \Theta(z,\tau \wedge \lambda)$. Takže žádný z takových literálů nebude mít vliv na velikost oblasti příslušející výslednému spojení. Stačí tedy uvažovat jen tyto výsledné klauzule $\tau$.

Krok 3 Algoritmu hledání spjatostního termu tedy nahradíme krokem 3', čímž získáme *algoritmus hledání termu odčítané oblasti.*

3'. $\forall l \in M(\tau)$ zkontrolujeme, zda je v $F$ nějaký větší term než $\tau \wedge l$.
Pokud ne, zkontrolujeme, zda nějaký vstupní bod $x_i \in \Theta(x, \tau \wedge l)$:

- pokud ano, $\tau$ vložíme do $F$ a z $F$ odstraníme všechny menší termy,
- pokud ne, $\tau \wedge l$ vložíme do $Q$, pokud už tam není.

1)  2) 

Obrázek 5.4: *1) Fuzzy objekt, který pojmul značnou část pozadí, 2) tentýž fuzzy objekt po odečtu pozadí. [6]*

Uvedený obrázek jasně ilustruje ideu metody. Obraz jsme segmentovali s výchozím bodem označeným červeně a prahem $\kappa(x,y) = [(3,0) \wedge (1,2) \wedge (2,1)] \vee [(2,1) \wedge (1,3)] \vee [(2,2)] \vee [(2,0) \wedge (1,4)] \vee (1,5) \vee [(4,0) \wedge (3,1)] \vee [(1,1) \wedge (8,0)] \vee [(6,0) \wedge (2,1)] \vee [(6,0) \wedge (1,3)]$. Výsledná oblast však obsahovala značnou část oblohy, proto jsme přistoupili k jejímu odečtení. Výchozím bodem pro odčítání se stal tmavě fialově označený bod a příslušný term byl $\tau = [(4,0) \wedge (1,1)] \vee [(2,0) \wedge (1,2)] \vee (1,3) \vee [(3,0) \wedge (2,1)]$. Odčítaná oblast je vyznačena svou hranicí ve fialové barvě.

Toto rozšíření je však prakticky použitelné jen v případě, že zahrnutý cizí objekt je poměrně jasně definován vůči nadobjektu vzniklého první segmentací. Odečítaný podobjekt musí vynikat vyšší spjatostí, než objekt původní.

# 6. Závěr

Ve druhé kapitole jsme si představili pojmy digitálního prostoru a digitálního obrazu. Seznámili jsme se s cíli a základními metodami prahování obrazu. Představili jsme si princip metody prahování a poukázali na jeho hlavní nedostatky, jako byla nulová navázanost výběru objektu na okolní hodnoty. Tento problém vylepšovala metoda fuzzy segmentace, která brala v úvahu podobnost sousedních pixelů. Seznámili jsme se s pojmem fuzzy spojitosti a s některými fuzzy objekty, ze kterých vycházela i následující metoda.

Představili jsme princip dalšího rozvoje metody, který směřoval k analýze více podobnostních znaků obrazu současně. Kvůli tomuto přístupu však bylo nutné vybudovat solidní algebraický podklad, který by podepřel celou metodu. Od uspořádaných množin jsme se dostali přes svazy k pojmu distributivity a volnosti svazu, vybudovali jsme příslušný volný distributivní svaz a poskytli několik rozdílných náhledů na tento pojem.

Pomocí prvků onoho svazu, volných termů, jsme představili vícekriteriální segmentaci. Vysvětlili jsme princip termového prahování a představili algoritmus získání vhodného termu. Zjistili jsme však, že se jedná obecně o výpočetně velmi složitý problém. I proto byl později ilustrován jen na poměrně malých obrazech.

Následná implementace ukázala, že i přes výpočetní náročnost lze dosáhnout jistých částečných výsledků. Představené možnosti urychlení se ukázaly jako výhodné a umožnily zpracování obrazu za zlomek původního času. I přesto však základní složitost algoritmu zůstává exponenciální.

Představená možnost rozšíření, která do segmentace vkládá pozitivní vliv lidského faktoru, umožňuje s minimálními zásahy zpřesnění hledané oblasti. Její výsledky jsou však pouze částečné.

# Literatura

[1] ADÁMEK, Jiří. *Matematické struktury a kategorie*. Vyd. 1. Praha: SNTL, 1982, 272 s.

[2] CARVALHO, Bruno M., C. Joe GAU, Gabor T. HERMAN, T. Yung KONG. Algorithms for Fuzzy Segmentation: Fuzzy Connectedness, Graph Cut and Related Algorithms. *Pattern Analysis & Applications*. 1999, **2**(1): 73-81. DOI: 10.4018/978-1-59904-889-5.ch073.

[3] DAVEY, B.A., PRIESTLEY, H.A. *Introduction to lattices and order*. 2nd ed. Cambridge: Cambridge University Press, 2002, xii, 298 s. ISBN 05-217-8451-4.

[4] GRATZER, George. *General lattice theory*. New York: Academic Press, 1978, xiii, 381 p. Pure and applied mathematics (Academic Press), 75. ISBN 01-229-5750-4

[5] HERMAN, Gabor T. *Geometry of digital spaces*. Boston: Birkhäuser, 1998, x, 216 s. ISBN 08-176-3897-0.

[6] PAVLÍK, Jan. Thresholding of a Digital Image by Free Terms. *Journal of Mathematical Imaging and Vision*. 2015, **51**(2): 338-354. DOI: 10.1007/s10851-014-0526-z. ISSN 0924-9907. Dostupné také z: http://link.springer.com/10.1007/s10851-014-0526-z

[7] ŠLAPAL, Josef. *Základy obecné algebry* [online]. Brno: Ústav matematiky FSI VUT v Brně, 2013 [cit. 2015-05-29]. Dostupné také z: http://www.math.fme.vutbr.cz/download.aspx?id_=3581

# A. Obsah přiloženého CD

Přiložený program vyžaduje kompilaci. Doporučené prostředí pro Windows je Qt Creator a framework Qt ve verzi 5.3 nebo vyšší spolu s kompilátorem Visual C++ 2013.

Pro Linux je nutný framework Qt ve verzi 5.3 nebo vyšší a kompilátor gcc verze 4.8 nebo vyšší. Zároveň doporučujeme použít prostředí Qt Creator.