Středoškolská Odborná činnost

Obor č. 1: Matematika a statistika

Příspěvek k popisu všech tříprvkových relačních struktur až na pp-konstruovatelnost

Jan Adam Zahálka
Ústecký kraj

Chomutov, 2022

Středoškolská Odborná činnost

Obor č. 1: Matematika a statistika

Příspěvek k popisu všech tříprvkových relačních struktur až na pp-konstruovatelnost

Toward characterizing all 3-element relational structures up to pp-constructibility

Autoři: Jan Adam Zahálka
Škola: Gymnázium Chomutov, p.o. Mostecká 3000, 430 01 Chomutov
Kraj: Ústecký kraj
Konzultant: doc. Mgr. Libor Barto, Ph.D.

Chomutov, 2022

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Chomutově dne <datum> ………………………………………………
Jan Adam Zahálka

Poděkování

Zde chci poděkovat Liboru Bartovi, za veškeré rady a asistenci, bez které by nebylo možné celou práci zrealizovat, Dmitriyi Zhukovi za poskytnutí vstupních dat a pomoc při jejich využití a mé přítelkyni Liubov za podporu.

Anotace

Tato práce má za cíl přiblížit se popisu všech tříprvkových relačních struktur definovatelných binárními a unárními relacemi až na pp-konstruovatelnost. Je známo že existuje 2 079 040 takových relačních struktur až na pp-definovatelnost. V této práci byl tento počet snížen na 1 207 relačních struktur, ze kterých jsou všechny ostatní pp-konstruovatelné.

Klíčová slova

CSP; relační struktury; klony; pp-konstruovatelnost

Annotation

The goal of this work was to get closer to characterizing all relational structures definable by binary and unary relations on three element set. It is known, that there exist 2 079 040 such relational structures up to pp-definability. In this work it was decreased to 1 207 relational structures, from which are all the other pp-constructible.

Keywords

CSP; relational structures; clones; pp-constructibility

Značení

- $\subseteq$ – relace býti podmnožinou
- $\mathbb{N}$ – množina přirozených čísel, tj. $\{1, 2, 3, ...\}$
- $|A|$ – kardinalita množiny A
- $P(A)$ – množina všech podmnožin množiny A
- A^B – množina všech zobrazení z B do A
- A^c – c-tá kartézská mocnina A
- $\mathrm{Op}(A)$ – množina všech operací na A
- $\mathrm{Op}_n(A)$ – množina všech n-árních operací na A
- $\mathrm{Rel}(A)$ – množina všech relací na A
- $\mathrm{Rel}_n(A)$ – množina všech n-árních relací na A
- $f\,[g_1, ..., g_k]$ – zobecněné složení operací f a g – Definice 2.1
- p_k^n – n-ární k-tá projekce – Definice 2.2
- $R(i_1, ...i_k)$ – projekce relace R na souřadnice $i_1, ..., i_k$ – Definice 2.6
- $\mathrm{Pol}(A)$ – množina všech funkcí zachovávajících všechny relace v A – Definice 2.7
- $\mathrm{Inv}(A)$ – množina všech invariantních relací pod všemi funkcemi z A – Definice 2.8
- $\overset{\text{hom.}}{\leq}$ – homomorfní uspořádání – Definice 3.5
- $\overset{\text{hom.}}{\equiv}$ – homomorfní ekvivalence – Definice 3.5
- $\overset{\text{hom.}}{\lesssim}$ – relace homomorfní ekvivalence druhému pp-poweru – Definice 3.12
- $\mathrm{Rel}(\mathbf{A})$ – množina všech relací v relační struktuře $\mathbf{A}$
- $\mathrm{Bin}(\mathbf{A})$ – množina všech binárních relací v relační struktuře $\mathbf{A}$
- $\mathrm{Un}(\mathbf{A})$ – množina unárních všech relací v relační struktuře $\mathbf{A}$
- $\mathbf{A}_0$ – relační struktura obsahující pouze relace $\{0\}, \{1\}, \{2\}$
- $\mathrm{S}(R)$ – symetrický uzávěr relace R – Definice 5.2
- $\mathrm{R}_A(R)$ – reflexivní uzávěr relace R do množiny A – Definice 5.2
- $\mathrm{T}(R)$ – tranzitivní uzávěr relace R – Definice 5.2

Obsah

1 Úvod **10**

2 Algebraické pozadí **12**
2.1 Základní definice 12
2.2 Korespondence polymorfismů a invariantních relací 14

3 CSP pozadí **16**
3.1 Základní definice a vztahy 16
3.2 Homomorfismy, pp-konstruovatelnost a majoritní operace 18

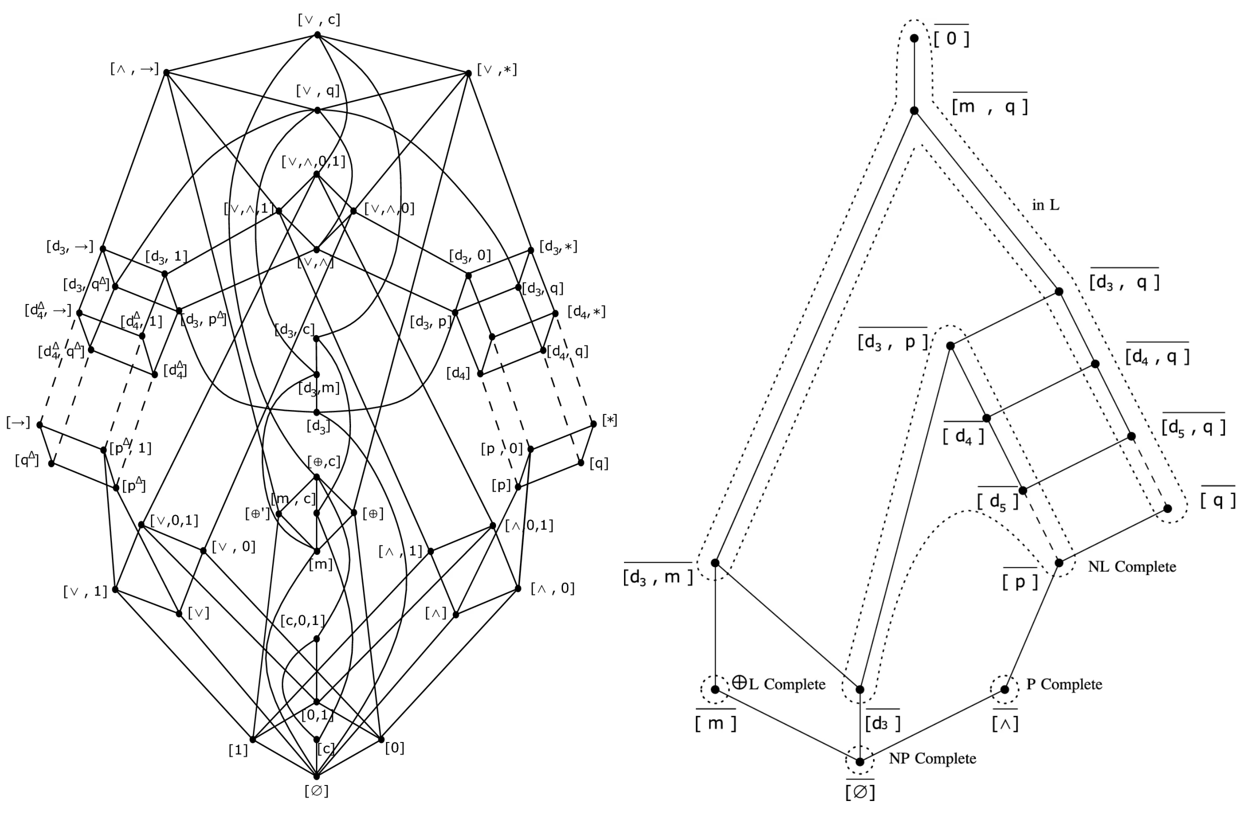
4 Návrh algoritmu **22**
4.1 Popis algoritmu 22
4.2 Zdrojový kód zajímavých částí algoritmu 23

5 Využití algoritmu a interpretace výsledků **26**

1 Úvod

V roce 1941 Emil Post zveřejnil kompletní svaz všech klonů[1] na dvouprvkové množině [12], který se ukázal jako relativně jednoduchý a jen spočetně nekonečný. Narozdíl od toho je podobný svaz na tříprvkové a nebo jiné větší množině je už nespočetný [11] a daleko složitější a tak se stává pro člověka do velké míry nedosažitelný. I přes komplikovanost tohoto svazu ale o něm existují nějaké výsledky. Všechny minimální klony[2] byly popsány v [8] a také i všechny maximální klony[3] tohoto svazu byly popsány v [13] a dokonce pro jeden ze zmíněných maximálních klonů jsou popsané (i přestože je jich také nespočetně mnoho) všechny podklony tohoto maximálního klonu [14].

Přirozeným postupem v takové situaci je nějakým způsobem redukovat množství klonů – vhodným způsobem takové redukce množství je pp-konstruovatelnost. Na dva klony budeme nahlížet jako na stejné pokud jsou v zápisu invariantními relacemi navzájem jeden z druhého pp-konstruovatelné. Při podobné redukci svazu všech klonů na tříprvkové množině opravdu získáme daleko "menší" svaz s jen spočetně mnoha prvky [6].



Obrázek 1: Svaz všech klonů na dvouprvkové množině (nalevo) a svaz všech klonů na dvouprvkové množině až na pp-konstruovatelnost (napravo) [5]

[1]Klony jsou algebraické struktury a budou definovány v kapitole 2.

[2]Klony, které jsou přímo nad nejmenším klonem, tj. klonem obsahujícím jen projekce

[3]Klony, které jsou přímo pod největším klonem, tj. klonem obsahujícím všechna zobrazení

Velkou motivací pro výzkum podobných svazů je Constraint Satisfaction Problem (zkráceně CSP), který se zabývá výpočetní složitostí problémů týkajících se otázky zda existují nějaké hodnoty splňující nějaká zadaná omezení. Mezi CSP problémy patří například známý SAT problém a nebo problém n-obarvitelnosti grafů. V určitém úhlu pohledu se na jednotlivé klony lze dívat jako na jednotlivé CSP problémy a svaz těchto klonů odpovídá zajímavým spojitostem, které mezi CSP problémy panují. Právě od teorie okolo CSP vychází výše zmíněná pp-konstruovatelnost.

Jedna z menších částí svazu klonů na tříprvkové množině je svaz všech klonů obsahujících majoritní funkci, který je konečný a byl celý popsán v [15] a s tím byl popsán i o trochu větší, ale stejnak konečný, nadsvaz tohoto svazu obsahující všechny klony definovatelné binárními relacemi, který je také konečný, ale velmi rozsáhlý – obsahuje 2 079 040 klonů. Zajímavá otázka je jak by takový klon vypadal až na pp-konstruovatelnost. Přiblížením se odpovědi na tuto otázku se bude zabývat tako práce.

2 Algebraické pozadí

Moderní matematika už dlouho směřuje k čím dál abstraktnějšímu chápání matematických objektů, například při studiu přirozených čísel si bylo potřeba uvědomit jen vlastnosti, která přirozená čísla mají a zavést tak abstraktnější objekt což v případě přirozených čísel byly okruhy a obory. Postupně ale lidé mířili ještě k daleko hlubší abstrakci a začaly nahlížet na podobné útvary jen jako na nějaké množiny spolu s nějakými operacemi (funkcemi s libovolným množstvím vstupů), dokonce i konstanty jsou z určitého úhlu pohledu operacemi s nulovým počtem vstupů (mají nulární aritu). Podobné struktury musí ale obsahovat veškeré i složené operace – ať máme například obor přirozených čísel $(\mathbb{N}, 0, 1, +, \cdot, -)$*, poté musíme zohlednit nejen operace jako* $a + b, c * d, -e$*, ale také* $(a * c) + b$ *nebo například* $((a * a * a) + (-c)) + 1$*, které se z těch základích operací skládají. Časem se ukázalo, že podobné soubory operací na množinách vystihují daleko více než jen čísla, ale také širokou škálu dalších objektů.*

Nejprve pro tuto sekci označme množinu všech operací na množině A, tj. $\bigcup_{n\in\mathbb{N}} A^{A^n}$ jako $\mathrm{Op}(A)$. Také označme množinu všech relací na množině A, tj. $\bigcup_{n\in\mathbb{N}} P(A^n)$ jako $\mathrm{Rel}(A)$. Také označme množinu všech n-árních operací na A, tj. A^{A^n} jako $\mathrm{Op}_n(A)$ a podobně i množinu všech n-árních relací na A jako $\mathrm{Rel}_n(A)$.

2.1 Základní definice

Definice 2.1 ([4])**.** Ať $n, k \in \mathbb{N}, f \in \mathrm{Op}_n(A), g_1, ..., g_n \in \mathrm{Op}_k(A)$. **Zobecněným složením** operace f s operacemi $g_1, ..., g_n$ nazveme k-ární operaci h na A definovanou jako

$$f\left[g_1, ..., g_k\right](x_1, ..., x_k) := f(g_1(x_1, ..., x_k), ..., g_n(x_1, ..., x_k))$$

[4]

Jak bylo zmíněno, tak skládání operací je pro nás jeden ze základních bodů konstrukce algebraických objektů. Narozdíl od běžného skládání funkcí je tato definice uzpůsobena i pro skládání operací arit vyšších než 1.

Definice 2.2 ([4])**.** Pro $n, k \in \mathbb{N}; k \leq n$ nazýváme n-ární k-tou **projekcí** n-ární operaci p_k^n takovou, že $p_k^n(x_1, ..., x_n) = x_k$.

Projekce nám slouží jako nástroj, který přijme libovolné n prvků a poté nám z nich vybere k-tý, což je velmi užitečné, pokud chceme dobře vystihnout skládání funkcí různých arit.

Definice 2.3 ([4])**.** Ať A je neprázdná množina. **Klonem** nad A nazveme množinu $\mathcal{K} \subseteq \mathrm{Op}(A)$, takovou, že obsahuje všechny projekce a je uzavřená na zobecněné skládání.

Klony jsou zcela zásadní pojem v této práci a do velké míry i v celé algebře. Klony nám popisují všechny "přípustné" operace pro nějakou množinu A. Klony lze také generovat

[4] $f\left[x_1, ..., x_k\right]$ je pouze symbolem, nejde o funkci která by brala za argumenty $g_1, ..., g_k$

z nějaké množiny operací způsobem, že najdeme nejmenší klon, který tyto operace obsahuje a tak poté je možné se na algebry dívat jen jako na generátory klonů [4].

Příklad 2.4. *Zde je několik příkladů klonů:*

1. $\mathrm{Op}(A)$ *je klonem nad* A
2. $\{p_k^n : 1 \leq k \leq n \in \mathbb{N}\}$*, tj. množina všech projekcí je klonem nad* A*.*
3. *Okruh polynomů nad* $\mathbb{Z}$ *je klonem nad* $\mathbb{Q}$

Konkrétně třetí část příkladu 2.4 dobře popisuje význam klonů. Je známo, že veškeré operace na $\mathbb{Q}$ lze zapsat do normálního tvaru, tj. tvaru

$$\sum c_j \cdot a_1^{n_1} \cdot a_2^{n_2} \cdot ... \cdot a_m^{n_m}; c_j, n_1, ..., n_m \in \mathbb{N}$$

neboli do součtu násobků součinů nějakých mocnin jednotlivých vstupních prvků a_i, což je ale přesně množina všech polynomů nad $\mathbb{Z}$.

Následující definice ukáže velmi podstatný vztah mezi operacemi a relacemi.

Definice 2.5 ([4])**.** Ať A je množina, $f \in \mathrm{Op}_n(A)$ a $\theta \in \mathrm{Rel}_k(A)$. Řekneme, že θ je **invariantní** vzhledem k f pokud

$$(x_{1,1}, x_{1,2}, ..., x_{1,k}), ..., (x_{n,1}, ..., x_{n,k}) \in \theta \Rightarrow (f(x_{1,1}, x_{2,1}, ..., x_{n,1}), ..., f(x_{1,k}, ..., x_{n,k})) \in \theta$$

Pokud θ je invariantní pod f, tak také říkáme, že f **zachovává** θ.

Jinak řečeno pokud zapíšeme prvky relace θ do tabulky takovým způsobem, že vždy řádek je prvek θ a poté s f zobrazíme jednotlivé buňky v řádku, tak nám vždy vyjde řádek, který je také prvkem θ.

$$\begin{array}{llllll}
x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,k} & \in \theta \\
x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,k} & \in \theta \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
x_{n,1} & x_{n,2} & x_{n,3} & \cdots & x_{n,k} & \in \theta \\
\downarrow f & \downarrow f & \downarrow f & & \downarrow f & \\
* & * & * & \cdots & * & \in \theta
\end{array}$$

Pokud na relaci nahlížíme jako na nějakou vlastnost mezi prvky A a pokud operace zachovává tuto relaci, tak skutečně tato vlastnost je poté zachována i mezi obrazy této operace.

Definice 2.6. Ať R je n-ární relace a $i_1, ...i_k \in \mathbb{N}, \forall j, 1 \leq i_j \leq n$. Poté definujme k-ární relaci

$$R(i_1, ...i_k) := \{(a_{i_1}, ..., a_{i_k}) : (a_1, ..., a_n) \in R\}$$

a nazývejme ji **projekce** R **na souřadnice** $i_1, ..., i_k$.

2.2 Korespondence polymorfismů a invariantních relací

Definice 2.7 ([4])**.** Pro libovolnou množinu relací Θ nad A definujme množinu $\mathrm{Pol}(\Theta) = \{f : A \to A \ : \ f$ zachovává $\theta, \forall \theta \in \Theta\}$. Tato množina je někdy nazývána "množina polymorfismů množiny Θ".

Definice 2.8 ([4])**.** Pro libovolné F, F je množinou funkcí z A do A definujme množinu relací $\mathrm{Inv}(F)$, takovou, že $\theta \in \mathrm{Inv}(F) \Leftrightarrow \forall f \in F$ je θ invariantní vzhledem k f.

Definice 2.7 a 2.8 vypadají velmi podobně a existuje mezi nimi velmi blízký vztah, který ukážeme v následujících větách.

Věta 2.9 ([9, 7])**.** *Ať Θ je množina relací nad $A \neq \emptyset$, poté* $\mathrm{Pol}(\Theta)$ *je klon.*

Důkaz. K důkazu stačí ověřit všechny podmínky klonu, tj.

1. Obsahuje všechny projekce

   Pro libovolnou projekci p_k^n a pro libovolnou $\theta \in \Theta$ vezměme libovolné $(x_{1,1}, ..., x_{1,k}), ..., (x_{n,1}, ...x_{n,k}) \in \theta$. Poté $(p_k^n(x_{1,1}, ..., x_{n,1}), ..., (x_{1,k}, ..., x_{n,k})) = (x_{k,1}, ..., x_{k,k})$, což je v θ a tj. p_k^n je v $\mathrm{Pol}(\Theta)$ pro všechna k, n. [5]

2. Je uzavřený na skládání

   Ať $f, g_1, ...g_m$ je libovolná funkce zachovávající θ a vezměme libovolné $\theta \in \Theta$ s aritou k a poté $(a_{i,1}, ..., a_{i,k}) \in \theta$ pro $i = 1, ..., k$. Podle předpokladu, že g_i zachovává θ platí $(g_i(a_{1,1}, ...a_{m,1}), ..., g_i(a_{1,k}, ...a_{m,k})) \in \theta$ pro všechna $i = 1, ..., m$. Poté protože f zachovává θ také platí

   $$(f\left(g_1(a_{1,1}, ..., a_{m,1}), ..., g_1(a_{1,k}, ..., a_{m,k})\right), ..., f\left(g_m(a_{1,1}, ..., a_{m,1}), ..., g_m(a_{1,k}, ..., a_{m,k})\right)) \in \theta$$

   Toto je ale rovno

   $$(f\left[g_1, ..., g_m\right](a_{1,1}, ..., a_{m,1}), ..., f\left[g_1, ..., g_m\right](a_{1,k}, ..., a_{m,k}))$$

   A to dokazuje, že také $f\left[g_1, ..., g_m\right]$ zachovává θ a tj. i $f\left[g_1, ..., g_k\right]$ je v $\mathrm{Pol}(\Theta)$.

□

Věta 2.10 ([9, 7])**.** *Ať $\mathcal{K}$ je klon nad konečnou množinou A. Poté* $\mathcal{K} = \mathrm{Pol}(\mathrm{Inv}(\mathcal{K}))$.

[5] Tento fakt je také velmi dobře viditelný z dříve zmíněné tabulky, kde f jen zobrazuje na jeden ze vstupních řádků.

Důkaz. Je zřejmé, že $\mathcal{K} \subseteq \mathrm{Pol}\,(\mathrm{Inv}(K))$ a proto se v tomto důkazu budeme zabývat jen opačnou inkluzí, tj. $\mathrm{Pol}\,(\mathrm{Inv}(K)) \subseteq \mathcal{K}$.

Pro libovolné $n \in \mathbb{N}$ seřaďme všechny prvky A^n do řady $\overline{c}_1, ..., \overline{c}_t$, kde $t = |A|^n$. Poté pro libovolné g n-ání definujme $\widehat{g} = (g(\overline{c}_1), ..., g(\overline{c}_t))$ a poté ať $\theta_n = \{\widehat{g} : g \in \mathcal{K}, g \text{ je } n\text{-ární}\}$ a $\Theta = \{\theta_n : n \in \mathbb{N}\}$.

Nyní ukažme, že $\mathrm{Pol}(\Theta) \subseteq \mathcal{K}$ a proto vyberme $f \in \mathrm{Pol}(\Theta)$ a označme jeho aritu jako n. Protože $f \in \mathrm{Pol}(\Theta)$, tak f zachovává θ_m pro všechna $m \in \mathbb{N}$, které mají aritu rovnou $t = |A|^m$, zvolme tedy $m = n$. Relace θ_n podle definice klonu obsahuje $\widehat{p_k^n} = (p_k^n(\overline{c}_1), ..., p_k^n(\overline{c}_t))$ pro všechna $k \leq n$. Protože f zachovává θ_n, tak $(f(p_1^n(\overline{c}_1), ..., p_n^n(\overline{c}_1)), ..., f(p_1^n(\overline{c}_t), ..., p_n^n(\overline{c}_t))) \in \theta_n$, ale $(f(p_1^n(\overline{c}_1), ..., p_n^n(\overline{c}_1)), ..., f(p_1^n(\overline{c}_t), ..., p_n^n(\overline{c}_t))) \in \theta_n$ se rovná $\widehat{f}$, z čehož vyplývá $f \in \mathcal{K}$.

Také $\Theta \subseteq \mathrm{Inv}(\mathcal{K})$, protože každé $\theta \in \Theta$ je inveriantní pod všemi $f \in \mathcal{K}$ z čehož také vyplývá $\mathrm{Pol}\,(\mathrm{Inv}(\mathcal{K})) \subseteq \mathrm{Pol}\,(\Theta)$. Poté z předchozího bodu a tranzitivity $\subseteq$ vyplývá $\mathrm{Pol}\,(\mathrm{Inv}(\mathcal{K})) \subseteq \mathcal{K}$.

□

Věty 2.9 a 2.10 nám společně říkají, že všechny klony lze zapsat jako $\mathrm{Pol}(\Theta)$ pro nějakou množinu relací Θ a zároveň $\mathrm{Pol}(\Theta)$ je klon pro libovolné Θ, neboli každému klonu odpovídá nějaká množina relací a každé množině relací nějaký klon. Obecně dokonce platí, že mezi klony a relacemi je takzvaná Galoisova korespondence – svazy množin relací a klonů na nějaké konečné množině A jsou navzájem izomorfní [4], ale tento poznatek zde nebude dokázán protože není nezbytný k této práci.

3 CSP pozadí

Zejména poslední dobou je velká tendence zkoumat které problémy jsou a které nejsou dobře výpočetně řešitelné. Například rozdělit nějaké číslo na součin prvočísel je algoritmicky velmi náročné, pro větší čísla dokonce v dnešní době prakticky nemožné. Jedním z problémů je také Constraint Satisfaction Problem, zkráceně CSP, který se zabývá zda nějaká relační struktura je splnitelná. Díky velké obecnosti definice tohoto problému do něj zapadá mnoho známých problémů jako třeba SAT, což je problém, zda je daná výroková formule splnitelná a nebo také například n-obarvitelnost grafu, což je problém, zda můžeme všechny vrcholy daného grafu obarvit n barvami tak, aby dva vrcholy stejné barvy nebyly spojeny hranou. V této kapitole projdeme definice, základní poznatky a motivace z tohoto oboru, které budou později využité v této práci.

3.1 Základní definice a vztahy

Základními objekty pro nás budou relační struktury. Relační struktury odpovídají jednotlivým CSP problémům, ale my na ně budeme nahlížet také z lehce jiné perspektivy.

Definice 3.1 ([2]). **Relační strukturou R** na konečné množině A s nazýváme dvojici $(A, (R_i : i \in I))$, kde $(R_i : i \in I$ je uspořádaná množina relací na A přes nějakou indexovou množinu I.

Pro relace a relační struktury používejme symbol relace náležení $\in$ tak, že $R \in \mathbf{R}$ platí právě tehdy když je R mezi relacemi $\mathbf{R}$. Také říkejme, že R je v $\mathbf{R}$.

Také pro relační strukturu $\mathbf{R}$ označme množinu $\mathrm{Rel}(\mathbf{R}) := \{R : R \in \mathbf{R}\}$.

První velmi podstatný koncept v teorii okolo CSP je pp-definovatelnost, pro kterou platí velmi zajímavý fakt – pokud je $\mathbf{R}$ pp-definovatelná z $\mathbf{S}$, tak poté lze $\mathbf{S}$ redukovat[6] na $\mathbf{R}$ (pokud na $\mathbf{R}, \mathbf{S}$ nahlížíme jako na CSP problémy) [2] – tento fakt zde ale nebude dokazován.

Definice 3.2 ([2]). Ať R je m-ární relace a $\mathbf{S}$ relační struktura obsahující relace $S_i : i \in I$. Relace R je **pp-definovatelná** z $\mathbf{S}$ pokud $R = \{(a_1, ..., a_m) : \varphi_{S_i:i\in I}(a_1, ..., a_m)\}$, kde $\varphi_{S_i:i\in I}(a_1, ..., a_m)$ je pozitivně primitivní formule, tj. skládá se jen z konjunkcí, existenčních kvantifikátorů, relace náležení k některé z S_i a rovnosti.

Také pro relační struktury $\mathbf{R}, \mathbf{S}$ řekněme, že $\mathbf{R}$ je **pp-definovatelná** z $\mathbf{S}$ pokud pro všechna $R \in \mathbf{R}$ je R pp-definovatelná z $\mathbf{S}$.

Příklad 3.3. *Ať*

$$R_1 = \{(0, 1), (2, 0), (1, 0)\}$$

[6]Redukce jednoho problému na jiný má v teorii výpočetní složitosti velký význam, protože prakticky znamená, že pokud se A redukuje na B, tak "B je jen speciálním případem A". Přesněji pojato to znamená, že existuje efektivní algoritmus, který problém A přeformuluje na problém B. Tato vlastnost také slouží k porovnávání obtížnosti problémů.

$$R_2 = \{(0,1),(1,2)\}$$
$$R_3 = \{(1,0),(2,0),(1,1)\}$$
$$S_1 = \{(0,1)\}$$
$$S_2 = \{(0,1,0),(0,2,0),(0,1,1)\}$$

a ať $\boldsymbol{R} = (\{0,1,2\},(R_1,R_2,R_3))$ a $\boldsymbol{S} = (\{0,1,2\},(S_1,S_2))$ Poté $\boldsymbol{S}$ je pp-definovatelná z $\boldsymbol{R}$, protože relaci S_1 lze definovat jako $\{(a,b) : (a,b) \in R_1 \wedge (a,b) \in R_2\}$ a relaci S_2 lze definovat jako $\{(a,b,c) : (\exists(n,m))(n,m) \in R_3 \wedge (a,n) \in R_1 \wedge (m,b) \in R_2 \wedge (b,c) \in R_3\}$.

Následující věta využije dříve zmíněných algebraických poznatků a ukáže podstatnou souvislost mezi klony relačních struktur a pp-definovatelností.

Věta 3.4 ([2]). *Ať $\boldsymbol{R}, \boldsymbol{S}$ jsou relační struktury nad množinou A, poté $\boldsymbol{R}$ pp-definuje $\boldsymbol{S}$ právě tehdy když $\mathrm{Pol}(\mathrm{Rel}(\boldsymbol{R})) \subseteq \mathrm{Pol}(\mathrm{Rel}(\boldsymbol{S}))$.*

Důkaz. Zde dokážeme pouze jednu implikaci – ta druhá je jednoduchá. Zde dokážeme, že pokud máme relační strukturu $\mathbf{R}$ a relaci S takovou, že je invariantní vzhledem ke všem $f \in \mathrm{Pol}(\mathrm{Rel}(\mathbf{R}))$, tak poté je S pp-definovatelná z $\mathbf{R}$, z čehož tvrzení ve formulaci věty vyplývá.

Předpokládejme $S(i,j) \nsubseteq \; =, \forall i,j \in \mathbb{N}$. Označme aritu S jako m a její kardinalitu jako n. Teď můžeme zapsat S ve formě matice typu $n \times m$, kde každý sloupec odpovídá jednomu prvku S, označme si tuto matici jako X. Matice X nemá díky předpokladu dva stejné řádky. Seřaďme všechny prvky A^n do řady $\overline{c}_1, ..., \overline{c}_t$, kde $t = |A|^n$. Ať g je operace s aritou n, poté ať $\widehat{g} = (g(\overline{c}_1), ..., g(\overline{c}_t))$.

Definujme relaci $\theta_n = \{\widehat{g} : g \text{ zachovává všechny relace v } \mathbf{R}\}$.

Relace θ_n je pp-definovatelná z $\mathbf{R}$ a to bez použití existenčních kvantifikátorů a rovnosti, tj. $(b_1, ..., b_t) \in \theta_n \Leftrightarrow \bigwedge \varphi : \varphi \in \phi$, kde ϕ obsahuje formule typu $(b_{i_1}, ..., b_{i_k}) \in R$, kde $i_1, ..., i_k \in \{1, ..., t\}$ pro k-ární $R \in \mathbf{R}$. Pro libovolné $R \in \mathbf{R}$ bude formule $(b_{i_1}, ..., b_{i_k}) \in R$ v ϕ právě tehdy, když $(\overline{c}_{i_1}(s), ..., \overline{c}_{i_k}(s)) \in R, \forall s \in \{1, ..., n\}$, kde $\overline{c}_i(k)$ označuje k-tou souřadnici $\overline{c}_i$.

Nyní definujme relaci S' takovou, že formuli definující θ_n, tj. $\bigwedge \varphi : \varphi \in \phi$ existenčně kvantifikujeme proměnnými $e_1, ..., e_l \in \{x \in A^n : x \text{ není řádek matice } X\}$, tj. nám vznikne m-ární relace S', protože z původní t-ární θ_n kvantifikujeme všechny až na m proměnných.

Nyní dokažme, že $S' = S$. Nejprve ukažme, že $S \subseteq S'$, tj. libovolné $\overline{s} \in S$ je také v S', což platí protože pro každou n-ární projekci p^n odpovídá $\widehat{p}^n$ jednomu prvku S. Inkluze $S' \subseteq S$ vychází z předpokladu, že S je invariantní vzhledem ke všem $f \in \mathrm{Pol}(\mathrm{Rel}(\mathbf{R}))$.

Nyní se zbavme předpokladu, tj. ať $\exists i,j : S(i,j) \subseteq \; =$. Označme si m jako aritu S a množinu $M = \{c : c = \max(\{i,j\}), S(i,j) \subseteq \; = \}$ a k její mohutnost. Ať $\overline{x}$ je $(m - |M|)$-tice souřadnic taková že neobsahuje souřadnice v M, tj.

$$\overline{x} = (1, ..., i_1 - 1, i_1 + 1, i_1 + 2, ..., i_k - 1, i_k + 1, ..., m); \forall j, 1 \leq j \leq k, i_j \in M$$

Definujme relaci $S' = S(\overline{x})$[7]. Relace S' splňuje předpoklad $S(i,j) \nsubseteq \; = , \forall i,j \in \mathbb{N}$ a tak víme že je pp-definovatelná z $\mathbf{R}$, tj. $S' = \{(a_x : x \in \overline{x}) : \phi(a_x, : x \in \overline{x})\}$ pro nějakou pozitivně primitivní ϕ. Relaci S poté můžeme definovat jako

$$S = \{(a_{x_1}, ..., a_{i_1}, ..., a_{i_k}, ...) : \phi(a_x, : x \in \overline{x}) \wedge \Big(\bigwedge\{c = i_j : 1 \leq j \leq k, S(c, i_j) \subseteq \; = \}\Big)\}$$

$\square$

Tato věta nám také říká, že pokud dvě relační struktury mají stejný příslušný klon, tak jsou navzájem pp-definovatelné.

3.2 Homomorfismy, pp-konstruovatelnost a majoritní operace

Dalším podstatným nástrojem na porovnávání relačních struktur jsou homomorfismy. Homomorfismy jsou zobrazení taková, že zachovávají strukturu jednotlivých relací, ale především mají také vlastnost, že pokud jsou $\mathbf{R}, \mathbf{S}$ relační struktury a jsou homomorfně ekvivalentní, tj. existuje homomorfismus z $\mathbf{R}$ do $\mathbf{S}$ a také existuje homomorfismus z $\mathbf{S}$ do $\mathbf{R}$, tak poté jsou $\mathbf{R}, \mathbf{S}$ na sebe navzájem redukovatelné (pokud na $\mathbf{R}, \mathbf{S}$ nahlížíme jako na CSP problémy) [2], ale tento poznatek tu také nebude dokazován.

Definice 3.5 ([2])**.** Ať R je relace na množině A a S je relace na množině B. Zobrazení $f : A \to B$ **zachovává pár relací** R a S pokud platí $\forall (a_1, ..., a_n) \in R : (f(a_1), ..., f(a_n)) \in S$.

Také mějme relační struktury $\mathbf{R} = (A, (R_i : i \in I)), \mathbf{S} = (B, (S_i : i \in I))$ takové, že R_i a S_i mají stejnou aritu pro všechna $i \in I$, poté funkci f nazveme **homomorfismem** z $\mathbf{R}$ do $\mathbf{S}$ pokud f zachovává pár relací R_i a S_i pro všechna $i \in I$. Fakt, že existuje homomorfismus z $\mathbf{R}$ do $\mathbf{S}$ značme $\mathbf{R} \overset{\text{hom.}}{\leq} \mathbf{S}$.

Pokud existuje homomorfismus z $\mathbf{R}$ do $\mathbf{S}$ a také z $\mathbf{S}$ do $\mathbf{R}$, tak relační struktury $\mathbf{R}, \mathbf{S}$ nazveme **homomorfně ekvivalentní**. Tento fakt značme $\mathbf{R} \overset{\text{hom.}}{\equiv} \mathbf{S}$.

Následující věta nám ukáže, že není potřeba dbát na pp-definovatelnost, pokud nás zajímá homomorfní ekvivalence nějakých relačních struktur.

Věta 3.6. *Ať $\boldsymbol{R} = (A, (R_i : i \in I)), \boldsymbol{S} = (B, (S_i : i \in I))$ jsou relační struktury. Poté $\boldsymbol{R} \overset{\text{hom.}}{\equiv} \boldsymbol{S}$ právě tehdy když pro libovolné R_j pro všechna $j \in J$ pp-definovatelné z $\boldsymbol{R}$ existují S_j pro všechna $j \in J$ pp-definovatelná z $\boldsymbol{S}$ takové, že $\boldsymbol{R} = (A, (R_i : i \in I, R_j : j \in J)) \overset{\text{hom.}}{\equiv} (B, (S_i \in I, S_j : j \in J))$.*

Důkaz. Zde dokážeme jen jednu implikaci, protože ta opačná je jednoduchá. Zde dokážeme že pokud $\mathbf{R} \overset{\text{hom.}}{\equiv} \mathbf{S}$, tak poté $\mathbf{R} = (A, (R_i : i \in I, R_j : j \in J)) \overset{\text{hom.}}{\equiv} (B, (S_i : i \in I, S_j : j \in J))$.

Pro tuto větu definujme pro libovolnou relaci R nad množinou A a libovolné $f : A \to B$ relaci $\overset{f\to}{R}$ jako $\{(f(a_1), ..., f(a_n)) : (a_1, ..., a_n) \in R\}$.

[7] Značením $S(\overline{x})$ je myšlena projekce S na souřadnice $\overline{x}$

Protože pro všechna $j \in J$ je R_j pp-definovatelná z $\mathbf{R}$, tak existuje formule $\varphi^j_{R_i:i\in I}$ taková že $R = \{(a_1, ..., a_n) : \varphi^j_{R_i:i\in I}(a_1, ..., a_n)\}$. Poté definujme formuli $\varphi^j_{S_i:i\in I}$ tak, že výskyt každého R_i nahradíme relací S_i. Poté definujme relaci $S_j = \{(a_1, ..., a_n) : \varphi^j_{S_i:i\in I}(a_1, ..., a_n)\}$. Fakt, že $\overset{h\rightarrow}{R_i} \subseteq S_i$ je zřejmě ekvivalentní tomu, že zobrazení h zachovává pár relací R_i, S_i. Označme si h_1 jako homomorfismus z $\mathbf{R}$ do $\mathbf{S}$. Poté $\overset{h_1\rightarrow}{R_i} \subseteq S_i$ pro všechna $i \in I$, což z definice R_j, S_j implikuje, že $\overset{h_1\rightarrow}{R_j} \subseteq S_j$ z čehož vyplývá, že h_1 zachovává pár relací R_j, S_j pro všechna $j \in J$ a z předpokladu této věty také, že zachovává pár relací R_i, S_i pro všechna $i \in I$, tj. h_1 je homomorfismus z $\mathbf{R} = (A, (R_i : i \in I, R_j : j \in J))$ do $(B, (S_i \in I, S_j : j \in J))$. Podobně lze ukázat že i opačný homomorfismus zachovává pár relací S_j, R_j pro všechna $j \in J$ a také pro S_i, R_i pro všechna $i \in I$. $\square$

Tato věta nám říká, že pro relační struktury můžeme přidat a případně i obebrat konečné množství pp-definovatelných relací bez toho, abychom změnili k nim příslušné homomorfismy, čehož později využijeme.

Homomorfismus z relační struktury na sebe samou nazýváme endomorfismem. Některé relační struktury mají příjemnou vlastnost, že na sobě mají jen "hezké" endomorfismy, tj. takové, že jsou bijektivní (poté se nazývají automorfismy) a některé dokonce ještě lepší vlastnost, že na nich existuje jen endomorfismus rovný identitě. V prvním případě tyto relační struktury označujeme jako "core", v druhém případě takové relační struktury označujeme jako "rigid core".

Definice 3.7 ([2])**.** Relační strukturu $\mathbf{R}$ nazýváme **core**, pokud každý endomorfismus, tj. homomorfismus z $\mathbf{R}$ do $\mathbf{R}$ je bijektivní.

Definice 3.8 ([3])**.** Relační strukturu $\mathbf{R}$ nazýváme **rigid core**, pokud každý endomorfismus, tj. homomorfismus z $\mathbf{R}$ do $\mathbf{R}$ je rovný identitě.

Hlavní důvod proč nás tako vlastnost zajímá je následné zjišťování homomorfní ekvivalence, protože pokud máme relační struktury $\mathbf{R}, \mathbf{S}$ rigid core, takové, že jsou homomorfně ekvivalentní, tak existuje nějaký homomorfismus f z $\mathbf{R}$ do $\mathbf{S}$ a také existuje nějaký homomorfismus g z $\mathbf{R}$ do $\mathbf{S}$ takový, že $g \circ f = id$ (toto je jednoduché pozorování z definic).

Další velmi užitečný pojem je pp-konstruovatelnost. Pp-konstruovatelnost má relativně podobný význam jako pp-definovatelnost, akorát v daleko větším měřítku. Pro pp-konstruovatelnost také platí vlastnost, že pokud $\mathbf{R}$ je pp-konstruovatelná z $\mathbf{S}$, tak poté je $\mathbf{R}$ redukovatelná na $\mathbf{S}$ (pokud vnímáme $\mathbf{R}, \mathbf{S}$ jako CSP problémy). Tento poznatek je také uveden bez důkazu.

Definice 3.9 ([2])**.** Ať S je k-ární relace na A^n, poté pro tuto definici označme

$$\widehat{S} := \{(a_{1,1}, a_{1,2}, ..., a_{k,n}) : ((a_{1,1}, ..., a_{1,n}), ...(a_{k,1}, ..., a_{k,n})) \in S\}$$

Ať $\mathbf{R}$ je relační struktura nad A a $\mathbf{S}$ je relační struktura nad A^n. Pro $m \in \mathbb{N}, 1 \leq n$ je $\mathbf{S}$ m**-tým pp-powerem** $\mathbf{R}$ pokud $\forall S \in \mathbf{S}$ je $\widehat{S}$ pp-definovatelná z $\mathbf{R}$.

Definice 3.10 ([2])**.** Ať $\mathbf{R}, \mathbf{S}$ jsou relační struktury. $\mathbf{R}$ je **pp-konstruovatelná** z $\mathbf{S}$ právě tehdy když $\mathbf{R}$ je homomorfně ekvivalentní S', kde S' je pp-power S.

Lze snadno ukázat, že relace pp-konstruovatelnosti je uspořádáním. Poté vzniklou ekvivalenci, tj. $\mathbf{R}$ je ekvivalentní $\mathbf{S}$ právě tehdy když $\mathbf{S}$ je pp-konstruovatelná z $\mathbf{R}$ a také $\mathbf{R}$ je pp-konstruovatelná z $\mathbf{S}$ nazývejme ekvivalencí pp-konstruovatelnosti.

Následující věta zmiňuje relativně zřejmý, ale podstatný vztah mezi pp-definovatelností a pp-konstruovatelností.

Věta 3.11 ([2])**.** *Ať $\boldsymbol{R}, \boldsymbol{S}$ jsou relační struktury nad množinou A. Jestliže $\boldsymbol{R}$ je pp-definovatelná z $\boldsymbol{S}$, tak $\boldsymbol{R}$ je pp-konstruovatelná z $\boldsymbol{S}$.*

Důkaz. Zvolme S' jako první pp-power S a všechny relace v S' definujme tak aby byly rovny relacím z $\mathbf{R}$, což jde protože je $\mathbf{R}$ pp-definovatelná z $\mathbf{S}$. $\square$

Následující definice definuje zásadní relaci v této práci. I když jde jen o velmi malou část pp-konstruovatelnosti, tak je to jedna z možností jak se pp-konstruovatelnosti přiblížit, tj. každé dva prvky co jsou v této relaci jsou i pp-konstruovatelné jeden z druhého (tato inkluze ale neplatí obráceně).

Definice 3.12. Definujme relaci $\overset{\text{hom.}}{\lesssim}$ takovou, že pro relační struktury $\mathbf{R}, \mathbf{S}$ platí $\mathbf{S} \overset{\text{hom.}}{\lesssim} \mathbf{R}$ právě tehdy když existuje $\mathbf{S}'$ druhý pp-power $\mathbf{S}$, takový, že $\mathbf{S}' \overset{\text{hom.}}{\equiv} \mathbf{R}$.

Následující věta nám ukáže, že pokud relační struktury, které jsou core vnímáme jako identické když jsou ze sebe navzájem pp-konstruovatelné, tak vždy existuje taková "forma" každé struktury aby byla rigid core.

Věta 3.13 ([3])**.** *Ať $\boldsymbol{R}$ je relační struktura, pak existuje $\boldsymbol{S}$ rigid core, taková, že $\boldsymbol{R}, \boldsymbol{S}$ jsou ze sebe navzájem pp-konstruovatelné.*

Důkaz. Pro každou relační strukturu $\mathbf{R}$ existuje $\mathbf{S}$ core, takový, že $\mathbf{R} \overset{\text{hom.}}{\equiv} \mathbf{S}$, což je dokázáno v [3].

Pro každý $\mathbf{S}$ core existuje $\mathbf{S}'$ rigid core takový, že $\mathbf{S}$ a $\mathbf{S}'$ jsou v ekvivalenci pp-definovatelnosti, což je dokázáno v [10].

$\square$

Některé relační struktury mají ve svém klonu zajímavé operace zvané majority, které umožní danou relační strukturu popisovat velmi jednoduše, s čímž nás seznámí následující definice a tvrzení.

Definice 3.14 ([2])**.** $f : A^3 \to A$ nazývejme majoritní operací (a nebo jen majoritou), pokud $\forall a, b \in A$ platí

$$f(a, a, b) = f(a, b, a) = f(b, a, a) = a$$

Také je vhodné poznamenat, že z definic přímo vyplývá, že tvrzení "Pol(Rel($\mathbf{R}$)) obsahuje majoritu" je stejné jako tvrzení "majorita zachovává všechna $R \in \mathbf{R}$" – toto tvrzení nadále také uvádějme jako "$\mathbf{R}$ je kompatibilní s majoritou".

Věta 3.15 ([1])**.** *Ať $\boldsymbol{R}$ je relační struktura kompatibilní s majoritou, pak lze každé $R \in \boldsymbol{R}$ definovat jen konjunkční formulí nad projekcemi R na všechny dvojice souřadnic , tj. $(a_1, ..., a_n) \in R \Leftrightarrow (a_{i_1}, a_{j_1}) \in R(i_1, j_1) \wedge ... \wedge (a_{i_k}, a_{j_k}) \in R(i_k, j_k)$, kde $(i_1, j_1), ...(i_k, j_k)$ jsou dvojice souřadnic, tj. $\forall (i_l, j_l), 1 \leq l \leq k; 1 \leq i_l \leq n, 1 \leq j_l \leq n, i_l \neq j_l$.*

Důkaz. Vyberme libovolné $R \in \mathbf{R}$ a označme jeho aritu jako n a poté vyberme jednu n-tici $(a_1, ..., a_n)$ v R. Dokažme indukcí, že $\forall I \subseteq \{1, ..., n\}, |I| \leq i; \exists (b_1, ...b_n) \in R; \forall i \in I, a_i = b_i$.

Pro $i = 2$ Existuje kvůli projekcím R na dvojice souřadnic.

Pro $i \geq 2$ označme $I = j_1, ...j_i$. Podle indukčního předpokladu známe $\bar{b}_1$ takové, že $a_i = b_{1_i}, \forall i \in I \setminus \{j_1\}$ a také podobné $\bar{b}_2$ a $\bar{b}_3$. Poté když na $\bar{b}_1, \bar{b}_2, \bar{b}_3$ provedeme po složkách majoritu (můžeme protože $\mathbf{R}$ je kompatibilní s majoritou), tak nám vyjde $\bar{b} \in R$ splňující $\forall i \in I, a_i = b_i$.

□

4 Návrh algoritmu

V této sekci bude popsán algoritmus, který rozhodne zda dvě relační struktury na množině $\{0, 1, 2\}$, které obsahují jen binární a unární relace a jsou kompatibilní s majoritou a jsou rigid core jsou v homomorfní relaci druhého pp-poweru.

Pro tuto sekci označme A jako množinu $\{0, 1, 2\}$ a $\mathbf{A} = (A, (R_i : i \in I)), \mathbf{B} = (A, (S_j : j \in J))$ jako dvě vstupní relační struktury a o kterých rozhodujeme, jestli $\mathbf{B} \overset{\text{hom.}}{\lesssim} \mathbf{A}$. Díky větě 3.6 můžeme zvolit $\mathbf{A}$ jako co nejmenší, tj. jen jako množinu relací takovou že žádná není z kterékoli jiné pp-definovatelná [8] a $\mathbf{B}$ jako co největší, tj. všechny binární a unární relace, které jsou z relací v $\mathbf{B}$ pp-definovatelné, bez toho abychom homomorfismy změnili.

4.1 Popis algoritmu

Tento algoritmus má za cíl najít relační strukturu $\mathbf{C}$ takovou, že $\mathbf{A} \overset{\text{hom.}}{\equiv} \mathbf{C}$ a zároveň $\mathbf{C}$ je druhý pp-power $\mathbf{B}$. O struktuře $\mathbf{C}$ víme, že je nad množinou A^2 a bude obsahovat binární a unární relace nad touto množinou A^2. Také víme, že všechna $h : A \to A$, která jsou homomorfní jsou rovna identitě[9] a právě toto budeme hledat jen mezi zobrazeními $f : A \to A^2, g : A^2 \to A$ takovými, že $g \circ f = id$.

1. Nejprve vyberme $f : A \to A^2$ takové, že f je injektivní [10].

2. Nyní musíme pro každou relaci v $\mathbf{A}$ vytvořit relaci v $\mathbf{C}$ stejné arity takovou, aby f bylo homomorfismus.

   Pro všechny $R_i : i \in I$ vytvořme odpovídající relaci $\overline{C}_i$. Pokud je R_i binární, tak $\overline{C}_i$ bude také binární a $\forall(a, b) \in R_i$ bude $(f(a), f(b)) \in \overline{C}_i$ a pokud R_i je unární, tak $\overline{C}_i$ je také unární a $\forall(a) \in R_i$ bude $(f(a)) \in \overline{C}_i$.

3. V tuto chvíli $\overline{C}_i$ obsahují jen nezbytné prvky k tomu aby f bylo homomorfismus z R_i do $\overline{C}_i$, ale nyní musíme najít nejmenší C_i takové, že obsahuje $\overline{C}_i$, aby f bylo homomorfismus a zároveň aby $\widehat{C}_i = \{(a_1, a_2, a_3, a_4) : ((a_1, a_2), (a_3, a_4)) \in C_i\}$ bylo pp-definovatelné z $\mathbf{B}$ (z definice pp-poweru).

4. Poté je ale také potřeba najít $g : A^2 \to A$ takové, že g je homomorfismus z $\mathbf{C}$ do $\mathbf{A}$ a také $g \circ f = id$. Protože $g \circ f = id$, tak víme, že g zobrazuje $f(0) \mapsto 0, f(1) \mapsto 1, f(2) \mapsto 2$. Pro každé $i \in I$ teď můžeme najít nějaké "vynucené" dvojice prvků a, b takové, že $g(a) = b$, tj. jestliže pro c takové, že víme, že $g(c') = c$ existuje právě jedna dvojice $(c, d) \in R_i$ a pokud existuje jednoznačné $g^{-1}(d)$ takové, že $g(g^{-1}(d)) = d$, tak poté musí $(c', g^{-1}(d)) \in C_i$, toto opakujeme dokud takové dvojice existují [11]. Pokud

[8] Množina takových relací se často nazývá jako množina generátorů $\mathbf{A}$

[9] Z předpokladu, že $\mathbf{A}$ je rigid core

[10] Jestliže hledáme bijektivní zobrazení $g \circ f$, tak musí být f injektivní

[11] Takové dvojice nemusí existovat žádné.

v tomto procesu dojde ke sporům, tj. g musí "vynuceně" mít dvě různé hodnoty pro stejný vstup, tak algoritmus rovnou rozhodne, že $\mathbf{B} \overset{\text{hom.}}{\not\lesssim} \mathbf{A}$.

5. Poté už zbývá jen projít prvky A^2 pro které g není definované a které se vyskytují v C_i a postupně jim přiřazovat různé obrazy a zjišťovat zda je poté g homomorfismus z $\mathbf{C}$ do $\mathbf{A}$. Pokud takové g existuje, tak $\mathbf{B} \overset{\text{hom.}}{\lesssim} \mathbf{A}$ a pokud ne tak $\mathbf{B} \overset{\text{hom.}}{\not\lesssim} \mathbf{A}$.

U některých kroků nemusí být zcela zřejmé jak je reálně provést. Vybrání f a poté vytvoření $\overline{C}_i$ je zřejmé, ale poté nalezení nejmenší pp-definovatelné C_i už tak zřejmé není. Nejprve si musíme uvědomit, že na pp-definici C_i stační kvůli větě 3.15 jen konjunkční formule, označme ji φ, ve které je jen konečně mnoho atomických formulí, označme je ψ_j, které mohou C_i "omezovat". Nyní pro každou ψ_j vyzkoušíme zda $\overline{C}_i \subseteq \{((a,b),(c,d)) : \psi_j(a,b,c,d)\}$ a pokud ano, tak se φ bude vyskytovat ve výsledné formuli φ a pokud ne, tak ji přeskočíme, tj.

$$\varphi = \bigwedge\{\psi_j : \overline{C}_i \subseteq \{((a,b),(c,d)) : \psi_j(a,b,c,d)\}\}$$

Poté stačí jen najít relaci $C_i = \{((a,b),(c,d)) : \varphi(a,b,c,d)\}$.

V dalším kroku hledáme "vynucené" části g, abychom zmírnili počet možností g, které je jinak poté procházet "hrubou silou", což je v exponenciálním čase vzhledem k množství nedefinovaných vstupů do g.

Při technickém pohledu nejde o příliš náročný algoritmus. V prvním kroku máme na výběr z 504 možností, druhý krok je výpočetně velmi jednoduchý, ve třetím kroku se pouze projdou všechny možné "omezení" které C_i ovlivňují, kde je 12 přípustných dvojic souřadnic pro celkové množství v případě binárních relací v $\mathbf{B}$ a rovnost, tj. zkoušíme zda je $\overline{C}_i$ splněna pro 4-ární relace C_i v případě binárních relací v B zkoušíme $12\cdot(|\operatorname{Bin}(B)|+1)$ a v případě unárních relací v B zkoušíme jen $4\cdot|\operatorname{Un}(B)|$ a pro binární relace C_i v případě binárních relací v B zkoušíme $2\cdot(|\operatorname{Bin}(B)|+1)$ a v případě unárních relací v B zkoušíme jen $2\cdot|\operatorname{Un}(B)|$. Při určování g maximálně $3^{|\operatorname{Bin}(A)|\cdot 2+\operatorname{Un}(A)-3}$ možností jak může být g definována. To je celkem přibližně $504\cdot(\operatorname{Bin}(A)\cdot(12\cdot(|\operatorname{Bin}(B)|+1)+4\cdot|\operatorname{Un}(B)|)+\operatorname{Un}(A)\cdot(2\cdot(|\operatorname{Bin}(B)|+1)+2\cdot|\operatorname{Un}(B)|))\cdot 3^{|\operatorname{Bin}(A)|\cdot 2+\operatorname{Un}(A)-3}$ kroků. Pro orientaci má nejmenší relační struktura (tj. ta co má nejvíce relací) 245 binárních relací a 6 unárních (tři binární relace generující všechny ostatní). V použité implementaci trvá tento proces přibližně čtvrtinu vteřiny.

4.2 Zdrojový kód zajímavých částí algoritmu

Celý program je psán v jazyce Python a zde nebudou definovány vestavěné funkce tohoto jazyka, které jsou ale velmi intuitivní a většinou sebevysvětlující.

V následující části kódu je popsaná funkce, která přijme list *SRel2* odpovídající množině binárních relací v $\mathbf{B}$ a list *SuRel2* odpovídající množině unárních relací v $\mathbf{B}$ a také relaci (ve formě "list listů") *VRel* odpovídající $\overline{C}_i$ a vrátí nejmenší C_i takovou, že obsahuje $\overline{C}_i$ a zároveň je pp-definovatelná z B.

```python
def získreltetra(SRel2,SuRel2,VRel):
    p = []
    #do tohoto listu budeme postupně přidávat "omezení" vzniklá z binárních relací v B
    for i in [[0,1],[0,2],[0,3],[1,0],[1,2],[1,3],[2,0],[2,1],[2,3],[3,0],[3,1],[3,2]]:
        for j in sjedn(SRel2,[[[0,0],[1,1],[2,2]]]):
        #toto je for pro všechny binární relace sjdenoceno s rovností
            if jepodmtetra(VRel,j,i):
            #jepodmtetra(R,S,i) vezme za vstup 4-ární relaci R, binární relaci S a dvojici souřadnic i a poté odpoví True pokud R(i_0,i_1) je podmnožina S a odpoví False pokud není
                p.append([j,i])
    p2 = []
    #do tohoto listu budeme postupně přidávat "omezení" vzniklá z unárních relací v B
    for i in [0,1,2,3]:
        for j in SuRel2:
            if jeunpodmtetra(VRel,j,i):
            #jeunpodmtetra(R,S,i) vezme za vstup 4-ární relaci R, unární relaci S a souřadnici i a poté odpoví True pokud R(i) je podmnožina S a odpoví False pokud není
                p2.append([j,i])

    e = []
    #do tohoto listu budeme postupně přidávat prvky A^4 splňující vše v listu p
    if len(p) != 0:
        for i in A4:
            z = True
            for j in p:
                dv = [i[j[1][0]],i[j[1][1]]]
                if not dv in j[0]:
                    z = False
                    break
            if z:
                e.append(i)
    else:
        e = copy.copy(A4)
    proz = []
    #do tohoto listu budeme postupně přidávat prvky e splňující vše v listu p2
    if len(p2) != 0:
```

```
        for i in e:
            z = True
            for j in p2:
                jed = [i[j[1]]]
                if not jed in j[0]:
                    z = False
                    break
            if z:
                proz.append(i)
        return(proz)
    else:
        return(e)
```

Následující část kódu obsahuje hledání zobrazení g ve čtvrtém kroku algoritmu popsaného výše. Jako vstup do této části programu jde množina 4-árních relací v $\mathbf{C}_i$ *SRel4*, binárních relací v $\mathbf{C}_i$ *SRel2*, množina binárních relací v $\mathbf{R}_i$ *SuRel2*, množina unárních relací v $\mathbf{R}_i$ *SuRel1* a zobrazení l odpovídající g s nějakými definovanými hodnotami. Před touto částí také je krátký inicializační program, který ale sám o sobě není moc zajímavý a tak zde nebude uveden.

```
while změna != 0:
        #proměnná do které se v každém cyklu ukládá počet zbylých nedefinovaných prvků
        změna = 0
        #proměnná do které se ukládá počet změn (nových definic)
        for s in range(len(SRel4)):
            Rel4 = SRel4[s]
            Rel2 = SRel2[s]
            for i in Rel4:
                if (dvěsouřv(i,l,0,1) == True) & (dvěsouřv(i,l,2,3) == False):
                #funkce dvěsouřv(R,S,i,j) přijímá jako vstup jednu 4-ární relaci R a odpoví zda R(i,j) je podmnožinou S
                        p = r(Rel2,f(l,[i[0],i[1]]))
                        #funkce r(R,c), kde R je binární vrátí list takový, že a ∈ list ⇔ (c,a) ∈ R a funkce f(F,c) vrátí F(c), kde F je definované způsobem list listů
                        if len(p) == 0:
                            return(None)
                        if len(p) == 1:
                            l.append([[i[2],i[3]],p[0]])
                            změna = změna +1
```

```
            if (dvěsouřv(i,l,0,1) == False) & (dvěsouřv(i,l,2,3) == True):
                p = robr(Rel2,f(l,[i[2],i[3]]))
                #funkce robr(R,c) funguje velmi podobně jako r(R,c), akorát obráceně, tj. vrátí list takový, že a ∈ list ⇔ (a,c) ∈ R
                if len(p) == 0:
                    return(None)
                if len(p) == 1:
                    l.append([[i[0],i[1]],p[0]])
                    změna = změna +1
    for i in range(len(SuRel2)):
        funk = f(l,SuRel2[i])
        if funk != None:
            if funk in SuRel1[i]:
                return(None)
```

Poté následuje program, který podle toho zda existují nedefinované hodnoty v l a nebo ne rozhodne zda začne zbylé nedefinované hodnoty doplňovat a nebo ne.

Umístění celého zdrojového kódu je sepsané ve volně vloženém listu u této práce.

5 Využití algoritmu a interpretace výsledků

Poté když máme možnost, jak algoritmicky zjistit jestli dvě relační struktury, které jsou kompatibilní s majoritou, jsou rigid core a jsou definovatelné binárními a unárními relacemi, tak můžeme pokročit dál a tento algoritmus využít k přiblížení se k popisu všech relačních struktur až na pp-konstruovatelnost.

K dispozici už je seznam všech relačních struktur s jen binárními a unárními relacemi až na pp-definovatelnost a je uspořádaný podmnožinovou inkluzí klonů těchto relačních struktur, tj. je to jen "obrácené" uspořádání pp-definovatelností. Tento seznam má ale několik problémů.

První velmi podstatný problém je, že relačních struktur v tomto seznamu je 2 079 040, takže pokud bychom zkoušeli všechny dvojice $\mathbf{R}, \mathbf{S}$, zda $\mathbf{R} \overset{\text{hom.}}{\lesssim} \mathbf{S}$ a také $\mathbf{S} \overset{\text{hom.}}{\lesssim} \mathbf{R}$ a orientačně bychom počátali s vyzkoušením čtyř relací za sekundu (což je s použitou implementací přibližně dobrý odhad) , tak by nám tato celá procedura trvala o něco méně než 35 000 let. Protože jde jenom o přiblížení se pp-konstruovatelnosti a ne její exaktní zjištění, tak zkusíme menší množinu dvojic relačních struktur a to právě ty dvojice, které jsou "přímo pod sebou" v uspořádání podle pp-konstruovatelnosti, tj. takové $\mathbf{R}, \mathbf{S}$, kde $\mathbf{R}$ je pp-definovatelná z $\mathbf{S}$ a zároveň neexistuje $\mathbf{C}, \mathbf{S} \neq \mathbf{C} \neq \mathbf{R}$ taková, že $\mathbf{R}$ je pp-definovatelná z $\mathbf{C}$ a také $\mathbf{C}$ je pp-definovatelná z $\mathbf{S}$. U takových dvojic kvůli větě 3.11 stační zkoušet tuto relaci jen jedním směrem a navíc takových dvojic je "jen" 9 737 075.

Další problém je, že ne všechny relační struktury, které obsahují jen binární a unární relace jsou také kompatibilní s majoritou, ale je jich převážná většina. Navíc pro dvě relační struktury výše zmíněný algoritmus nikdy neřekne, že dvě relační struktury by byly v této relaci i když by v ní být neměli, přinejhorším akorát řekne, že v relaci druhého pp-poweru nejsou i když by být mohli, což nám ale příliš nevadí, protože jde jen o přiblížení se pp--konstruovatelnosti.

Poslední už ne tak závažný problém je, že všechny struktury v tomto seznamu jsou core, ale ne všechny jsou rigid core. Tento problém je ale snadno řešitelný díky větě 3.13, která nám říká, že pro každý core existuje i rigid core takový, že je v ekvivalenci pp-konstruovatelnosti. Ten najdeme způsobem, že najdeme největší rigid core a poté i všechny pod ním budou rigid core, největší rigid core v tomto svazu je relační struktura obsahující pouze relace $\{0\}, \{1\}, \{2\}$, označme ji jako $\mathbf{A}_0$ (na pořadí relací nezáleží protože jsou všechny navzájem pp-definovatelné).

Věta 5.1. *Pro všechny $\boldsymbol{R} \in V$ rigid core je $\boldsymbol{A}_0$ pp-definovatelná z $\boldsymbol{R}$.*

Důkaz. Proveďme důkaz sporem a předpokládejme, že existuje $\mathbf{R}$ rigid core takový, že $\mathbf{A}_0$ není pp-definovatelná z $\mathbf{R}$. Tento předpoklad je stejný jako, že existuje klon $\mathcal{K}$ takový, že $\forall f \in K, f$ nezachovává $\{0\}, \{1\}, \{2\}$, tj. $\exists c \in \{0, 1, 2\}, \exists g \in K, g$ nezachovává $\{c\}$. Protože $\{c\}$ obsahuje pouze jeden prvek, tak z faktu, že g nezachovává $\{c\}$ plyne $g(c, ..., c) \neq c$. Operace $g\left[p_1^1, ..., p_1^1\right]$ je také v $\mathcal{K}$, pro kterou platí $g\left[p_1^1, ..., p_1^1\right](c) \neq c$, což je ve sporu s faktem, že $\mathbf{R}$ je rigid core. $\square$

Výše zmíněný algoritmus je lepší vnímat jako kdyby rozhodoval jestli dvě relační struktury jsou pp-konstruovatelné jedna z druhé a nebo nevíme zda jsou a nebo ne, než nějaké exaktní pravidlo.

Pro tuto kapitolu si označme graf $G = (V, E)$, kde V je množina všech rigid core (získáme je tím, že z původního listu vybereme jen ty relační struktury, které jsou pod A_0). definovatených binárními relacemi až na pp-definovatelnost a E jsou všechny dvojice relačních struktur, které jsou "přímo pod sebou", tj. takové $(\mathbf{R}, \mathbf{S})$, kde $\mathbf{S}$ je pp-definovatelná z $\mathbf{R}$ (tj. $\mathrm{Pol}(\mathrm{Rel}(\mathbf{R})) \subseteq \mathrm{Pol}(\mathrm{Rel}(\mathbf{S}))$) a zároveň neexistuje $\mathbf{C}$ taková, že $\mathbf{R}$ je pp-definovatelná z $\mathbf{C}$ a také $\mathbf{C}$ je pp-definovatelná z $\mathbf{S}$.

Pro všechny hrany $(\mathbf{R}, \mathbf{S})$ bylo vyzkoušeno zda $\mathbf{R} \overset{\text{hom.}}{\lesssim} \mathbf{S}$. Pokud bylo zjištěno, že pro $(\mathbf{R}, \mathbf{S})$ platí $\mathbf{R} \overset{\text{hom.}}{\lesssim} \mathbf{S}$, tak poté víme, že $\mathbf{R}$ a $\mathbf{S}$ jsou ze sebe navzájem pp-konstruovatelné – relaci takových $(\mathbf{R}, \mathbf{S})$ označme jako C. Protože ekvivalence pp-konstruovatelnosti je ekvivalencí, tak je potřeba udělat nejdříve reflexivní a symetrický uzávěr a poté také tranzitivní uzávěr, tyto pojmy budou definovány v následujících definicích.

Definice 5.2. Ať $\sim$ je relace a A libovolná množina, poté ať

1. $\mathrm{S}(\sim) = \{(a, b) : a \sim b \vee \ b \sim a\}$
2. $\mathrm{R}_A(\sim) = \{(a, a) : a \in A\} \cup \sim$
3. $\mathrm{T}(\sim) = \{(a, b) : \exists(a, a_1, ..., a_n, b); a \sim a_1 \wedge a_1 \sim a_2 \wedge ... \wedge a_n \sim b\}$

S($\sim$) nazýváme symetrický uzávěr, $\mathrm{R}_A(\sim)$ reflexivní uzávěr a T($\sim$) tranzitivní uzávěr relace $\sim$.

Označme si ekvivalenci $\equiv$ jako $\mathrm{T}(\mathrm{R}_V(\mathrm{S}(C)))$, pro kterou zřejmě platí, že pokud $\mathbf{R} \equiv \mathbf{S}$, tak poté je $\mathbf{R}$ v ekvivalenci pp-konstruovatelnosti s $\mathbf{S}$ (obrácená inkluze neplatí).

Poté si označme nový graf $G' = (V', E')$, kde V' obsahuje všechny třídy ekvivalence $\equiv$ a E' obsahuje všechny dvojice $([\mathbf{R}], [\mathbf{S}])$[12] takové, že $\exists \mathbf{A} \in [\mathbf{R}], \exists \mathbf{B} \in [\mathbf{S}], (\mathbf{A}, \mathbf{B}) \in E$.

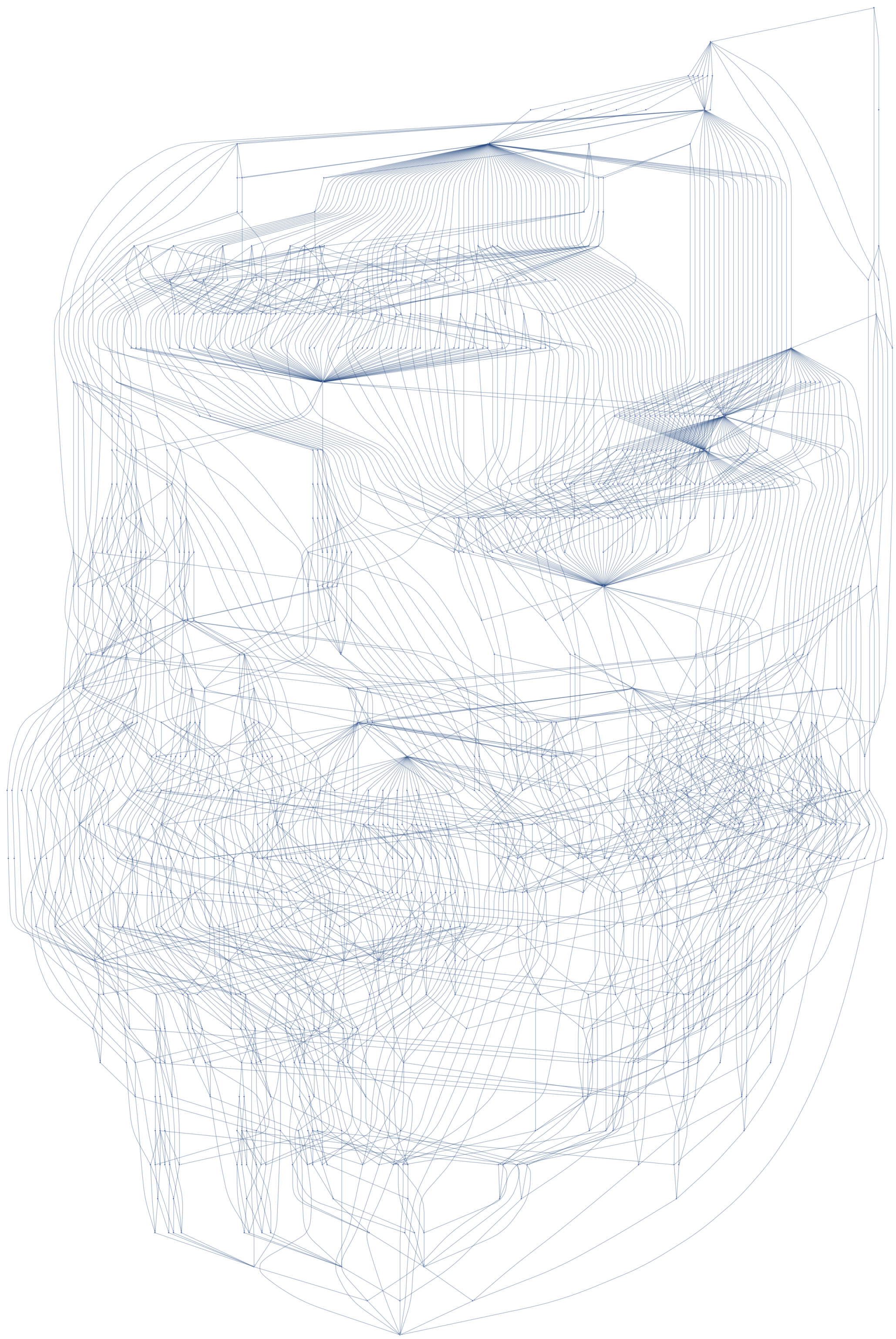
Věta 5.3. *Pro všechny cykly $\overline{c}$ v G' platí, že všechny relační struktury $\boldsymbol{R} \in [\boldsymbol{C}] \in \overline{c}$ jsou po dvou v ekvivalenci pp-konstruovatelnosti.*

Důkaz. Ať $\overline{c} = (c_1, ...c_n)$ je nějaký cyklus v G', tj. $(c_i, c_{i+1}) \in E', \forall i < n, (c_n, c_1) \in E'$. To, že nějaké $(a, b) \in E'$, pokud znamená, že $\exists \mathbf{R} \in a, \exists \mathbf{S} \in b$ takové, že $\mathbf{S}$ je pp-definovatelná z $\mathbf{R}$ a také pro dvě relační struktury v jedné třídě platí, že jsou v relaci pp-konstruovatelnosti, tj. $\mathbf{R}, \mathbf{S} \in c_i, i \leq n$ platí, že $\mathbf{R}, \mathbf{S}$ jsou v ekvivalenci pp-konstruovatelnosti. Pro libovolné $c_i, c_j \in \overline{c}$, bez újmy na obecnosti $i < j \leq n$ (pokud $i = j$, tak je to zřejmé) ať $\mathbf{R} \in c_i, \mathbf{S} \in c_j$. Fakt, že $\mathbf{S}$ je pp-konstruovatelná z $\mathbf{R}$ je zřejmý z poznatků výše, protože existuje řada $c_i, c_{i+1}, ..., c_j$, kde je libovolná relační struktura z libovolného prvku této řady je pp-konstruovatelná z libovolné relační struktury toho předchozího a obráceně $\mathbf{R}$ je pp-konstruovatelná z $\mathbf{S}$ protože existuje řada $c_j, c_{j+1}, ..., c_n, c_1, ...c_i$, kde je také libovolná relační struktura z libovolného prvku pp-konstruovatelná z libovolné relační struktury toho předchozího. □

Věta výše nám ukazuje, že je také potřeba se zaobírat cykly v grafu G', protože poté pokud existují nějaké cykly (které existují) můžeme množství tříd ještě změnšit, resp. vytvořit relaci $\simeq$ takovou, že $\equiv \subseteq \mathrm{T}(\mathrm{R}_V(\mathrm{S}(\simeq)))$ a zároveň takovou, že implikuje ekvivalenci pp-konstruovatelnosti. Tuto relaci nalezneme jako $\simeq = \equiv \cup \{(a, b) : \exists \text{ cyklus } \overline{c}; a, b \in \overline{c}\}$.

Vytvořme graf $G'' = (V'', E'')$ takový, že V'' obsahuje všechny třídy ekvivalence $\mathrm{T}(\mathrm{R}_V(\mathrm{S}(\simeq)))$ a E'' definujme podobně jako E'. Z původních 2079040 vrcholů původního grafu prvcích nám vznikne 1 207 tříd, kde nejmohutnější obsahuje 1 329 769 relačních struktur. Mediánem počtu relačních struktur ve třídě je 3, což je zajímavé protože jsou relační struktury mezi třídy rozděleny velmi nerovnoměrně (5 největších tříd dohromady obsahuje přibližně 95% veškerých relačních struktur z V). Vrcholy mají průměrně přibližně 2, 2 vrcholů "přímo pod nimi". Na obrázku níže je grafické znázornění grafu G''.

[12] $[a]$ označuje třídu ekvivalence obsahující prvek a



Obrázek 2: Graf G''

Umístění veškerých souborů obsahujících popisy jednotlivých popsaných objektů v tomto postupu a jejich popisy jsou sepsané ve volně vloženém listu u této práce.

Reference

[1] Kirby A. Baker and Alden F. Pixley. Polynomial interpolation and the Chinese remainder theorem for algebraic systems. *Mathematische Zeitschrift*, 143:165--174, 1975.

[2] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Andrei Krokhin and Stanislav Živný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1--44. Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017.

[3] Libor Barto, Jakub Opršal, and Michael Pinsker. The wonderland of reflections. *Israel J. Math.*, 223(1):363--398, 2018.

[4] C. H. Bergman. *Universal algebra: Fundamentals and selected topics*. CRC Press, 2012.

[5] Manuel Bodirsky and Albert Vucaj. Two-element structures modulo primitive positive constructability. *Algebra Universalis*, 81(2):Paper No. 20, 17, 2020.

[6] Manuel Bodirsky, Albert Vucaj, and Dmitriy Zhuk. The lattice of clones of self-dual operations collapsed, 2021.

[7] V. G. Bodnarchuk, L. A. Kaluzhnin, Viktor N. Kotov, and Boris A. Romov. Galois theory for Post algebras. I. *Cybernetics*, 5(3):243--252, 1969.

[8] Béla Csákány. Minimal clones—a minicourse. *Algebra Universalis*, 54:73--89, 2005.

[9] David Geiger. Closed systems of functions and predicates. *Pacific J. Math.*, 27:95--100, 1968.

[10] Pavol Hell and Jaroslav Nešetřil. *Graphs and homomorphisms*, volume 28 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2004.

[11] Ju. I. Janov and A. A. Mučnik. Existence of k-valued closed classes without a finite basis. *Dokl. Akad. Nauk SSSR*, 127:44--46, 1959.

[12] Emil L. Post. *The Two-Valued Iterative Systems of Mathematical Logic*. Annals of Mathematics Studies, No. 5. Princeton University Press, Princeton, N. J., 1941.

[13] S. V. Yablonskiĭ. On functional completeness in a three-valued calculus. *Doklady Akad. Nauk SSSR (N.S.)*, 95:1153--1155, 1954.

[14] Dmitriy Zhuk. The lattice of the clones of self-dual functions in three-valued logic. In *2011 41st IEEE International Symposium on Multiple-Valued Logic*, pages 193--197, 2011.

[15] Dmitriy Zhuk and Stanislav Moiseev. On the clones containing a near-unanimity function. In *2013 IEEE 43rd International Symposium on Multiple-Valued Logic*, 2013.

Soubory

Data jsou uložena do souborů za pomoci knihovny Pickle (viz dokumentace ZDE). Všechny soubory jsou ke stažení na webové stránce ZDE pod nadpisem "Files" ve složce "Příspěvek k popisu všech tříprvkových relačních struktur až na pp-konstruovatelnost".

Na převod relací z úsporného do standardního formátu (tj. list listů) lze použít funkcí *GetBinaryRelations* a *GetUnaryRelations*.

- *lattice.b* – list obsahující veškeré relační struktury definovatelné binárními a unárními relacemi, kde každá je zapsaná formátem listu délky 7, kde
    1. index relační struktury
    2. (zbytné) index vrstvy
    3. (zbytné) umístění
    4. binární relace, které tato struktura obsahuje (v úsporném formátu)
    5. unární relace, které tato struktura obsahuje (v úsporném formátu)
    6. (zbytné) počet maximálních podklonů
    7. list indexů maximálních podklonů
- *lattice-konst.b* – list obsahující veškeré rigid core definovatelné binárními a unárními relacemi ve stejném formátu jako *lattice.b*
- *lattice-konst-usp.b* – list obsahující listy délky 2 popisující uspořádání mezi třídami z *lattice-konst.b* (obsahuje jen maximální dvojice, tj. ty dvojice kde na druhé souřadnici je klon a na první je jeho maximální podklon)
- *lattice-dict-lw.b* – slovník odkazující indexy relačních struktur na list délky dva obsahující na nulté souřadnici binární relace v této relační struktuře (v úsporném formátu) a na první souřadnici unární relace v této relační struktuře (v úsporném formátu).
- *lattice-gen-dict-lw.b* – slovník odkazující indexy relačních struktur na list délky dva obsahující na nulté souřadnici generující binární relace v této relační struktuře (v úsporném formátu) a na první souřadnici generující unární relace v této relační struktuře (v úsporném formátu).[1]
- *arch.b* – list stejné délky jako *usb.b*, kde každá souřadnice na každé souřadnici je buď $False$ a nebo důkaz že je odpovídající dvojice v *usp.b* v relaci homomorfní ekvivalence druhého pp-poweru. Důkazy jsou listy délky 4, kde
    1. homomorfismus $\mathbf{A}$ do $\mathbf{C}$[2]

[1] "Generujícími" je myšleno, že všechny relace v této struktuře jsou pp-definovatelné z binárních a unárních relací příslušné relační struktury

[2] $\mathbf{A}$ je relační struktura která má být homomorfně ekvivalentní $\mathbf{C}$, kde $\mathbf{C}$ je druhý pp-power $\mathbf{B}$

2. homomorfismus z **C** do **A**
3. 4-ární relace v **C**
4. binární relace v **C**

- *arch-lw.b* – podobné jako *arch.b*, akorát obsahuje jen odpověď *True* a nebo *False* bez důkazu.
- *popisclas.b* – list obsahující všechny třídy ekvivalence $\simeq$ ve formě listů délky 7, kde
  1. index třídy (indexy byly přidělovány podle počtu prvků které daná třída obsahuje, tj. index 0 má ta třída která obsahuje nejvíce klonů, index 1 ta třída která obsahuje nejvíce klonů kromě té první atp.)
  2. indexy klonů, které tato třída obsahuje
  3. Nic.
  4. Nic.
  5. Nic.
  6. (zbytné) Počet maximálních podtříd
  7. list maximálních podtříd (viz kapitola 5 v psané práci)
- *source.py* – kompletní zdrojový kód použitých skriptů