UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA

BAKALÁŘSKÁ PRÁCE

Vlastnosti konečných grup s programem GAP

Katedra algebry a geometrie
Vedoucí bakalářské práce: **Doc. RNDr. Petr Emanovský, Ph.D.**
Vypracoval(a): **Tereza Nedjalková**
Studijní program: B1101 Matematika
Studijní obor Matematika, Informatika se zaměřením na vzdělávání
Forma studia: prezenční
Rok odevzdání: 2018

BIBLIOGRAFICKÁ IDENTIFIKACE

Autor: Tereza Nedjalková

Název práce: Vlastnosti konečných grup s programem GAP

Typ práce: Bakalářská práce

Pracoviště: Katedra algebry a geometrie

Vedoucí práce: Doc. RNDr. Petr Emanovský, Ph.D.

Rok obhajoby práce: 2018

Abstrakt: Tato práce si klade za cíl seznámit čtenáře se základy teorie grup a dále blíže s konečnými grupami a jejich vlastnostmi. Ty jsou demonstrovány na příkladech jak řešených, tak pouze zadaných k procvičení studenty samotnými. Příklady jsou řešitelné v programu GAP.

Klíčová slova: GAP, teorie grup, algebra, konečnost

Počet stran: 61

Počet příloh: 0

Jazyk: český

BIBLIOGRAPHICAL IDENTIFICATION

Author: Tereza Nedjalková

Title: Properties of finite groups with the program GAP

Type of thesis: Bachelor's

Department: Department of Algebra and Geometry

Supervisor: Doc. RNDr. Petr Emanovský, Ph.D.

The year of presentation: 2018

Abstract: This bachelor's thesis aims to acquaint the reader with the basics of group theory and further closer to the finite groups and their properties. These are demonstrated on solved examples. Part of the text are unresolved examples for practise. Exercises are designed for solving in GAP system.

Key words: GAP, group theory, algebra, finite

Number of pages: 61

Number of appendices: 0

Language: Czech

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně pod vedením pana Doc. RNDr. Petra Emanovského, Ph.D. a všechny použité zdroje jsem uvedla v seznamu literatury.

V Olomouci dne ....

podpis

Obsah

Seznam obrázků

Úvod

"For example"is not proof.

Židovské přísloví

Pro svou bakalářskou práci jsem si zvolila téma: "Vlastnosti konečných grup s programem GAP". Cílem práce je sepsat stručný manuál k matematickému softwaru GAP a následně jeho funkčnost ukázat na příkladech zaměřených na vyšetřování vlastností konečných grup.
Hlavním důvodem pro zvolení daného tématu je skutečnost, že mým studijním oborem je kombinace matematiky a informatiky se zaměřením na vzdělávání. Můžu tedy ve své práci zužitkovat znalosti z obou oblastí studia.
Práce by měla sloužit jako pomůcka pro ty, kteří chtějí své znalosti teorie grup aplikovat v matematickém softwaru GAP a zjistit, že spousta zajímavých vlastností, které však vyžadují složitější výpočty se dá ověřit během pár minut.
Pro přehlednost jsou v práci shrnuty základní pojmy, definice a věty, které čtenář při řešení příkladů využije. Text je tak přístupný i pro ty, kteří se v teorii grup zatím příliš neorientují a chtějí si rozšířit obzory i v základních kategoriích.

Tato práce je rozdělena do šesti kapitol, které jsou na konci doplněny o cvičení, které jsou sestavena tak, aby si čtenář sám procvičil nabyté vědomosti. Řešení ke cvičením je uvedeno na konci práce. V první kapitole s názvem *Úvod do programu GAP* je čtenář seznámen se základní syntaxí programu, aby v něm byl schopen pracovat i člověk, jež se s programováním zatím nesetkal. Je tedy první částí manuálu k programu.

Ten pokračuje v druhé kapitole *Úvodní pojmy,* kde je již doplněn o základní pojmy abstraktní algebry, bez kterých bychom se v konečných grupách jen těžko obešli.

Následuje kapitola *Úvod do teorie grup*, která obsahuje formální definici grupy takovou, jak ji známe z hodin algebry. V kapitole je uvedena řada příkladů různých grup, které si můžeme pomocí GAP ukázat v praxi. V této kapitole se také poprvé seznamujeme s pojmem konečná grupa a to na příkladech grup symetrických, které jsou speciální podskupinou konečných grup.

Čtvrtá kapitola nese název *Permutační grupy* a podává nám největší vysvětlení ohledně značení a symboliky při práci s GAP, jelikož v programu jsou grupy reprezentovány permutacemi. V páté kapitole, *Konečné grupy, podgrupy* si shrneme vlastnosti konečných grup jako takových. Může se zdát, že ač práce nese název *Vlastnosti konečných grup s programem GAP*, je zvláštní, že kapitola s "konečnými grupami"v názvu je až v druhé polovině textu. V práci jsem však zaměřila dost prostoru i již zmíněným grupám symetrií, které taktéž patří mezi konečné grupy. Kapitola 5 tedy shrnuje základní vlastnosti, se kterými jsme se již mohli setkat v předchozích kapitolách, spolu s těmi, které jsme v textu doposud neuvedli. Zavádí také pojem podgrupy, který je v problematice konečných grup stěžejní.

Kapitola *Cyklické grupy* je poslední kapitolou mé bakalářské práce. Shrnuji v ní základní vlastnosti cyklických grup, které jsou taktéž nedílnou součástí teorie konečných grup. Cyklické grupy jsou svými vztahy mezi řády prvků a řády grup samotných velmi zajímavé a proto v mé práci nesmí chybět.

Po uvedených šesti kapitolách přikládám část nesoucí název *Řešení úloh*, která, jak název napovídá, obsahuje výsledky příkladů, které byly v textu uvedeny pro čtenářovo samostatné řešení. V práci uvádím přehledně základní věty potřebné k dané problematice. Důkazy k těmto větám jsou dostupné v publikaci Contemporary abstract algebra (GALLIAN, Joseph A.). [2]

Kapitola 1

Úvod do programu GAP

Tato kapitola obsahuje základní pokyny pro používání bezplatného softwaru GAP. Tento manuál předpokládá použití verze 4.8.8, která je volně stažitelná na adrese `http://www.gap-system.org/`. Referenční příručka a rozsáhlý manuál pro software GAP jsou k dispozici na téže stránkách.
Některé příkazy, které budeme často používat:

- Pro ukončení GAP zadejte příkaz *quit*;
- Použijte šipky nahoru/dolů pro opětovné zobrazení předchozích příkazů
- Pokud chyba způsobí zacyklení (a zobrazí výzvu $brk >$), můžete ukončit smyčku zadáním příkazu $< ctl > -D$ nebo *quit*;
- Zadejte ? následovaný názvem tématu pro pomoc s tímto tématem.

Poznámka. *Výše uvedený příkaz ? je obzvláště užitečný. Může být použit vždy, když zapomenete přesný příkaz, nebo když nevíte, zda v GAP pro konkrétní úkon příkaz existuje.*
Předpokládejme, že chcete vědět, jak naznačit násobení v GAP.
Zadejte příkaz gap > ?multiplication a GAP vrátí seznam příkazů pro násobení různých algebraických struktur.

Poznámka. *POZOR:*

- *GAP rozlišuje mezi velkými a malými písmeny*
- *Na konci příkazu vždy přidejte středník*
- *Dva středníky na konci příkazu způsobí vyhodnocení příkazu, ale výsledek nebude vypsán*

1.1. Základní funkce GAP

Nyní si na cvičení ukážeme, jak prostředí GAP vypadá.
Do příkazového řádku napište $(5+3)*9$; a stiskněte Enter. Vaše obrazovka by měla vypadat takto:

```
gap> (5+3)*9;
72
```

Nyní do příkazového řádku napište výraz $(5+3)*9$ (bez středníku) a stiskněte Enter.

```
gap> (5+3)*9
>
```

Všimněme si, že se nic nestane. GAP neví, že jsme příkaz dokončili, protože chybí středník. Dopíšeme-li středník, bude nám vrácena hodnota 72.

```
gap> (5+3)*9
>;
72
gap>
```

Nyní zadejte $(5+3*9$; a stiskněte Enter. Na obrazovce byste měli mít následující výraz.

```
gap> (5+3*9;
Syntax error: ) expected
```

Došlo k chybovému hlášení, protože nesouhlasí počet levých a pravých závorek v příkazovém řádku. Použijte šipku nahoru pro opětovné zobrazení předchozího příkazu. Potom použijte klávesy se šipkami pro doplnění potřebné závorky.

```
gap> (5+3*9;
Syntax error: ) expected
gap> (5+3)*9;
72
gap>
```

GAP lze též použít k otestování rovnosti dvou hodnot. Do příkazového řádku napište $6 = 9$; a stiskněte klávesu Enter.

```
gap> 6=9;
false
```

GAP vrátil hodnotu false, protože 6 není rovno 9.

Příkaz přiřazení v GAP

Hodnotu proměnné v GAP přiřadíme pomocí :=. Umožňuje nám odkazovat na objekt pomocí pojmenování.

Název proměnné se nazývá *identifikátor*

V následující příkladu je a identifikátor:

```
gap> a:= (10+7)*(9-6);
51
gap> a;
51
gap> a*(a-1);
2550
gap> a:=14 ;;
gap> a*(a-1);
182
```

Všimněte si, že při přidělení identifikátoru je jeho hodnota vypsána na dalším řádku. Protože se v řádku $a := 14;;$ vyskytly dva středníky, hodnota a nebyla vypsána, i když byla změněna.

Téměř jakákoli sekvence písmen a číslic, obsahující alespoň jedno písmeno, může být použita jako identifikátor.

Cvičení

1. Základní funkce GAP

   (a) Vypočtěte hodnotu 3^{121}

   (b) Zjistěte, zda $2^{25} + (45 * 51777)$ je větší, než 34 milionů;

2. Příkaz přiřazení v GAP

   (a) Přiřaďte hodnotu 14 identifikátoru b

   (b) Přiřaďte hodnotu 123456789 identifikátoru $BigNUMBER$

   (c) Vypočtěte $b + BigNUMBER$

Kapitola 2

Úvodní pojmy

2.1. Vlastnosti celých čísel

Velkou část abstraktní algebry tvoří vlastnosti celých čísel a množin. V této kapitole shrneme nejdůležitější vlastnosti, které budeme v následujícím textu využívat.
Dalším zásadním pojmem v teorii čísel je pojem *dělitelnost*.
Řekneme, že nenulové číslo t je *dělitelem* čísla s právě tehdy, když existuje číslo u, pro které platí $s = tu$. V tomto případě píšeme $t \mid s$ (čteme t *dělí* s). Jestliže t není dělitelem s, píšeme $t \nmid s$.
Říkáme, že číslo s je násobkem čísla t právě tehdy, když existuje u takové, že $s = ut$, nebo analogicky, když $t \mid s$. *Prvočíslo* je kladné číslo, jehož jediní kladní dělitelé jsou 1 a číslo samo.

Věta 1 (Celočíselné dělení se zbytkem)**.** *Nechť jsou a, b celá čísla, kde $b > 0$. Pak existuje právě jedno celé číslo p a právě jedno r tak, že $a = bq + r$, kde $0 \leq r < b$.*

Číslo q v dělícím algoritmu nazveme *celočíselný podíl*, číslo r se nazývá *zbytek po dělení*.

Příklad 2.1.1. Pro $a = 17$ a $b = 5$, dělící algoritmus dává $17 = 5 \cdot 3 + 2$; pro $a = -23$ a $b = 6$ dostáváme $-23 = 6(-4) + 1$.

Největším společným dělitelem nenulových čísel a a b nazveme největší ze všech společných dělitelů a, b. V tomto textu budeme toto číslo označovat $gcd(a, b)$ (Greatest Common Divisor).
V případě, že $gcd(a, b) = 1$, řekneme, že čísla a a b jsou *nesoudělná*.
Nejmenší společný násobek dvou nenulových čísel a a b je nejmenší kladné číslo, které je násobkem čísla a a současně čísla b. V tomto textu budeme toto číslo označovat $lcm(a, b)$ (Least Common Multiple).

Vlastnosti celých čísel v GAP

Software GAP obsahuje spoustu předdefinovaných funkcí. Funkce v GAP začínají velkým písmenem.
Například funkce *Gcd* (*Lcm*) pro výpočet největšího společného dělitele (nejmenšího společného násobku) celých, nenulových čísel. Příklady:

```
gap> Gcd(123, 456);
3
gap> Lcm(123,456);
18696
```

Věta 2. *Pro všechna nenulová celá čísla a a b existují celá čísla s, t taková, že největší společný dělitel prvků a, b je roven $as + bt$.*

Pomocí funkce *Gcdex* vypočítáme čísla s,t z předchozí věty.

Příklad 2.1.2.
```
gap> Gcdex(4,15);
rec(coeff1:= 4, coeff2:= -1, coeff3:= -15, coeff4:= 4,
 gcd:= 1)
gap>
```

Výše uvedený postup nám říká, že největším společným dělitelem čísel ($gcd(4, 15)$) je číslo 1 a $Gcd(4, 15) = 4 * 4 + (-1) * 15 = 1$. To znamená, že $coeff1$ a $coeff2$ jsou celá čísla s,t taková, že $Gcd(a, b) = as + bt$. Výstupy $coeff3$ a $coeff4$ jsou celá čísla m,n, pro která platí, že $am + bn = 0$.

2.2. Modulární aritmetika

Další aplikace dělících algoritmů, která pro nás bude důležitá, se nazývá modulární aritmetika.

Modulární aritmetika je metodou počítání, které každý z nás jistě někdy použil. Například, jestliže nyní je duben, jaký měsíc bude za 25 měsíců? Samozřejmě, že každého ihned napadne květen i bez toho, aniž by odpočítal 25 měsíců od dubna, protože víme, že $25 = 2 \cdot 12 + 1$ a stačilo tedy přidat jeden měsíc k dubnu. Převedeme-li tento příklad do matematické symboliky, můžeme zapsat, že $25 \mod 12 = 1$.

Můžeme tedy zapsat, že jestliže $a = qn + r$, kde q je podíl a r je zbytek po dělení a číslem n, zapíšeme tuto skutečnost jako $a \mod n = r$.

Modulární aritmetika v GAP

GAP umí také pracovat s modulární aritmetikou.

Příklad:

```
gap> 23 mod 6;
5
gap> 5 mod 6 + 11 mod 6;
10
gap> 10 mod 6;
4
gap> (5 mod 6 + 11 mod 6) mod 6;
4
```

Poznámka. *POZOR: funkce mod je funkční pouze se správně umístěnými mezerami. Výraz* $23mod6$ *by byl v GAP vnímán, jako identifikátor.*

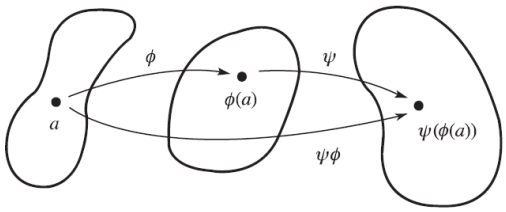
2.3. Funkce (zobrazení)

Funkce hrají velkou roli téměř ve všech směrech matematiky, taktéž terminologie a značení s funkcemi spojeno. V následující kapitole si shrneme pojmy potřebné v tomto textu.

Funkce (zobrazení) ϕ z množiny A do množiny b je pravidlo, které každému prvku a z množiny A přiřazuje právě jeden prvek b z množiny B. Množina A se nazývá *definiční obor* ϕ a množina B *obor hodnot* ϕ.

Jestliže se ve zobrazení ϕ prvek a zobrazí na prvek b, pak prvek a nazveme *vzor* a b se nazývá *obraz prvku* a ve ϕ. Podmnožina množiny B obsahující všechny obrazy prvků z množiny A se nazývá *obraz množiny* A *ve* ϕ.

Pro skutečnost, že ϕ je zobrazení z množiny A do množiny B užíváme zápis $\phi : A \longrightarrow B$. Pro zápis zobrazení prvku b z B na prvek a z A užijeme značení $\phi(a) = b, a \in A, b \in B$, nebo $\phi : a \longrightarrow b$ a říkáme, že a se zobrazuje na b. Mějme zobrazení $\phi : A \longrightarrow B$ a $\psi : B \longrightarrow C$. *Složením zobrazení* ϕ, ψ je zobrazení z množiny A do množiny C definováno jako $(\phi\psi)(a) = \psi(\phi(a)) \forall a \in A$. Složení si můžeme znázornit následujícím obrázkem.

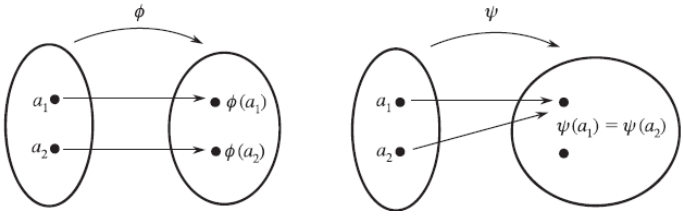


Obrázek 2.1: Složené zobrazení $\phi\psi$

V početních úlohách bývá složení dvou funkcí f, g znázorněno $(f \circ g)(x)$ a je definováno následovně: $(f \circ g)(x) = f(g(x))$.

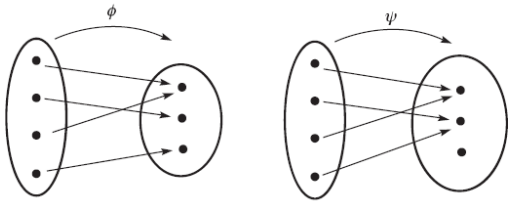
Příklad 2.3.1. Nechť $f(x) = 2x + 3$ a $g(x) = x^2 + 1$.
Pak $(f \circ g)(5) = f(g(5)) = f(23) = 55$; $(g \circ f)(5) = g(f(5)) = g(13) = 170$;
Obecně tedy $(fg)(x) = 2(x^2 + 1) + 3 = 2x^2 + 5$,
$(gf)(x) = (2x + 3)^2 + 1 = 4x^2 + 12x + 10$.
Tedy je vidět, že skládání zobrazení není komutativní. $((fg)(x) \neq (gf)(x))$

Zobrazení $\phi : A \longrightarrow B$ se nazývá *injektivní* (prosté zobrazení), právě když pro všechna $a_1, a_2 \in A$ platí, že jestliže se rovnají jejich obrazy, rovnají se i vzory.
Symbolicky: $\phi(a_1) = \phi(a_2) \Rightarrow a_1 = a_2$.



Obrázek 2.2: Injektivní × neinjektivní zobrazení

Zobrazení $\phi : A \longrightarrow B$ se nazývá *surjektivní* (zobrazení "na"), právě když každý prvek b z množiny B je obrazem alespoň jednoho prvku a z množiny A.
Symbolicky: $\forall b \in B \exists a \in A : \phi(a) = b$.



Obrázek 2.3: Surjektivní × nesurjektivní zobrazení

Zobrazení, které je injektivní i surjektivní současně, nazveme *bijektivní zobrazení.*

Věta 3 (Vlastnosti zobrazení)**.** *Jsou dána zobrazení α, β, γ, pro která platí:*

$$\alpha : A \longrightarrow B, \beta : B \longrightarrow C, \gamma : C \longrightarrow D$$

1. $\gamma(\beta\alpha) = (\gamma\beta)\alpha$ *(asociativita);*
2. *Jsou-li α a β injektivní, pak složení $\beta\alpha$ je injektivní;*
3. *Jsou-li α a β surjektivní, pak složení $\beta\alpha$ je surjektivní;*
4. *Je-li α bijektivní, pak existuje inverzní zobrazení $\alpha^{-1} : B \longrightarrow A$ takové, že $\forall a \in A, \forall b \in B : (\alpha^{-1}\alpha)(a) = a$ a $(\alpha^{-1}\alpha)(b) = b$.*

Funkce - *mapping* v GAP

Pomocí GAP můžete také vytvářet vlastní funkce. Jedním ze způsobů je použití operátoru *maps-to*, označeného $->$.
V následujícím příkladu si vytvoříme funkci *square*, která přijímá jedno číslo a vrací hodnotu jeho druhé mocniny.

```
gap> square:=x -> x^2;
function( x ) ... end
```

Nyní si funkčnost ověříme na příkladu.

```
gap> square(2);
4
gap> square(5);
25
```

Zkuste do GAP vložit následující příkaz.

```
gap> 6162256386 mod 11;
0
```

Program ověřil, že číslo 6162256386 je dělitelné 11 beze zbytku. To mimo jiné znamená, že by to mohlo být rodné číslo osoby žijící v ČR, protože jedním z kontrolních znaků rodných čísel je právě dělitelnost 11. Zkuste si tedy zadat například

```
gap> 6162286386 mod 11;
3
```

a vidíme, že tento sled číslic by za rodné číslo být považován nemohl.

Jiným způsobem ověření dělitelnosti určitou hodnotou je také

```
gap> 6162256386 mod 11 = 0;
true
gap> 6162286386 mod 11 = 0;
false
```

Cvičení

6 1 6 2 2 5 / 6 3 8 6
61: ročník narození
62: měsíc narození (u žen +50)
25: den narození
638: "pořadové číslo" narození pro dané datum
6: hodnota kontrolní číslice

Obrázek 2.4: Způsob generování rodného čísla

1. Vlastnosti celých čísel

   (a) Vypočítejte největšího společného dělitele a nejmenší společný násobek:

      i. čísel 25646 a 68185

      ii. čísel 9245 a 325

   (b) Použijte funkci *Gcdex* k nalezení koeficientů s, t pro které platí, že $gcd(8701, 10057) = 8701s + 10057t$

2. Funkce

   (a) Vytvořte funkci, která přijímá jako argument kladné celé číslo n a vrací hodnotu součtu $1 + 2 + \cdots + n$. Poté využijte tuto funkci pro $n = 6$ a $n = 2584$.

   (b) Vytvořte funkci, která přijímá jako argument libovolné celé číslo t a vrací hodnotu

   $$f(t) = t^3 + 3 - 2t$$

   (c) Vytvořte funkci, která ověří validitu zadaného rodného čísla;

      i. použijte Vámi vytvořenou funkci pro zkontrolování Vašeho rodného čísla;

      ii. změňte v zadaném čísle jednu číslici. Dokázala Vámi napsaná funkce rozeznat chybné zadání?

Kapitola 3

Úvod do teorie grup

> Symmetry is a vast subject, significant in art and nature. Mathematics lies at its root, and it would be hard to find a better one on which to demonstrate the working of the mathematical intellect.
>
> Hermann Weyl, Symmetry (1952)

3.1. Pojem grupa

Pojem *grupa* byl poprvé použit Évariste Galoisem okolo roku 1830 pro popis množin injektivních zobrazení na konečných množinách, které by mohly být seskupeny dohromady a vytvořily by tak strukturu uzavřenou na skládání.
Jako v případě většiny nejzákladnějších pojmů v matematice, i definice grupy kterou známe dnes a kterou si zde uvedeme je výsledkem dlouhého evolučního procesu. Moderní definici grupy podal Walther von Dyck v roce 1882.

Definice 1. Nechť G je množina. *Binární operací na* G rozumíme zobrazení, které přiřazuje každé uspořádané dvojici prvků z G jednoznačně daný prvek z G.

Binární operace na množině je tedy metoda, která spojením dvou prvků z množiny G určuje jiný prvek množiny G. Podmínka, že výsledek operace je prvek množiny G nazveme *uzavřenost.* Nejznámějšími binárními operacemi jsou sčítání, odečítání či násobení celých čísel.

Dělení celých čísel není binární operací na množině celých čísel, protože podílem dvou celých čísel nemusí být výhradně celé číslo.

Binární operace *sčítání* mod n a *násobení* mod n na množině $\{0, 1, ..., n-1\}$, které značíme Z_n hraje velmi důležitou roli v abstraktní algebře. V některých případech budeme uvažovat množinu Z_n pouze s operací sčítání mod n, jindy budeme chtít využít jak sčítání mod n, tak násobení mod n. Například pro násobení matic sestavených z prvků množiny Z_n potřebujeme jak násobení mod n, tak sčítání mod n.

Definice 2. Nechť G je množina uvažována spolu s binární operací, která přiřazuje každé uspořádané dvojici $(a, b) \in G^2$ prvek množiny G, který budeme značit ab[1]. Řekneme, že struktura $G = (G, \cdot)$ je *grupa*, právě když platí následující podmínky:

1. *asociativita:* Operace je asociativní, tedy

$$\forall a, b, c \in G : (ab)c = a(bc);$$

2. *existence jednotkového prvku:* existuje prvek $e \in G$ takový, že platí

$$\forall a \in G : ae = ea = a;$$

3. *existence inverzních prvků:* pro všechny prvky a množiny G existuje v G prvek a^{-1} takový, že platí

$$aa^{-1} = a^{-1}a = e.$$

Jestliže pro všechny prvky a, b grupy G platí $ab = ba$, říkáme, že grupa je *Abelova* (komutativní). Grupa není Abelova, existuje-li v ní alespoň jedna dvojice prvků, pro kterou komutativita neplatí ($ab \neq ba$).

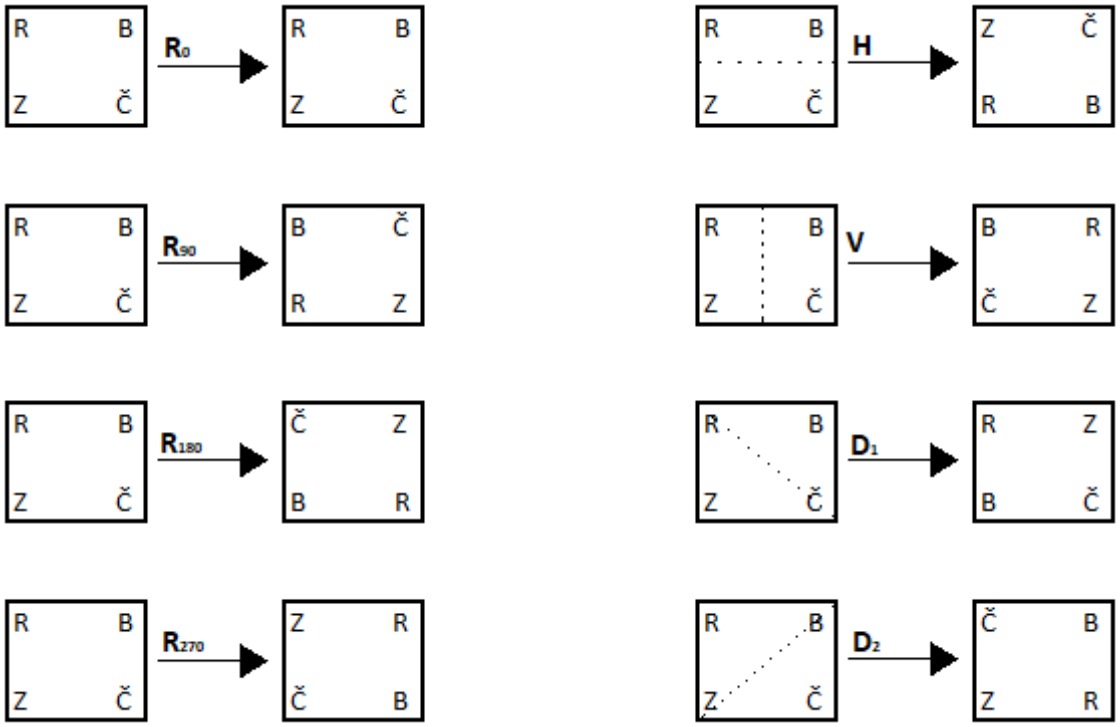
[1] G^2 symbolizuje kartézský součin $G \times G$; $G^2 = \{(a, b) | a, b \in G\}$

3.2. Symetrie čtverce

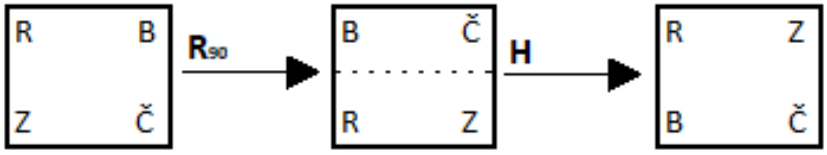
Předpokládejme, že z roviny odebereme čtvercovou oblast, nějakým způsobem ji přesuneme a vložíme čtverec zpátky do prostoru, kde byl původně. V této kapitole popíšeme všechny možné způsoby, jak to lze udělat.
Konkrétněji řečeno, chceme popsat možné vztahy mezi počáteční a konečnou polohou čtverce z hlediska pohybu. Zajímá nás však více čistý dopad pohybu na polohu čtverce, než pohyb samotný.
Například rotaci o 90° a rotaci o 450° považujeme za rovnocenné, protože mají stejný důsledek pro každý bod čtverce. Tímto zjednodušením je dosažení našeho cíle o poznání jednodušší.
Pro začátek můžeme uvažovat čtvercovou oblast s rohy označenými různými barvami - černá (Č), bílá (B), růžová (R) a zelená (Z). Díky tomuto označení lze snadno rozlišit pohyby, které se v účinku na čtverec vzájemně liší.
S tímto schématem jsme nyní schopni jednoduše popsat všechny možné způsoby, jak lze přesunout náš čtvercový objekt a ukážeme si to na následujícím obrázku, kde R_α značí otočení (rotaci) o úhel α, H překlopení po horizontální ose, V překlopení po vertikální ose, D_1 překlopení po hlavní diagonále a D_2 překlopení po vedlejší diagonále.

Dalším obrázkem si ukážeme, že jakýmkoli složením daných symetrií získáme polohu, jež máme definovanou výše, tedy že ve čtverci žádné další symetrie neexistují.

Vezmeme-li v potaz značení, které jsme si zavedli v předchozí kapitole, zapíšeme tuto skutečnost jako $HR_{90} = D_1$. Množinu uvedených osmi symetrií spolu s operací skládání symetrií nazveme *dihedrální grupa řádu* 8



Obrázek 3.1: Symetrie čtverce



Obrázek 3.2: $R_0 \to R_{90} \to H = D_1$

Abychom byli schopni považovat strukturu za grupu, je nutno ověřit, zda operace skládání zobrazení na množině symetrií má požadované vlastnosti, tedy je-li asociativní a zda v ní existuje neutrální prvek a také ke každému prvku grupy prvek inverzní:

1. *asociativita:* operací dané struktury je skládání zobrazení, jež je asociativní;
2. *existence neutrálního prvku:* neutrálním prvkem je tzv. identita (R_0), tedy zobrazení, které ponechává objekt na svém místě;
3. *existence inverzních prvků:* vzhledem k bijektivitě symetrií, ke každé symetrii musí existovat příslušné inverzní zobrazení, které nazveme symetrie inverzní[2].

3.3. Dihedrální grupy (D_n)

Jak jsme již nastínili v úvodu kapitoly, dihedrální grupa řádu $2n$ je grupou symetrií pravidelného n-úhelníka, kde množinou symetrií rozumíme kromě identity osovou souměrnost a rotaci. Dihedrální grupy jsou jednoduchým příkladem konečných grup, které jsou hlavním předmětem mé práce.
Dihedrální grupy se často objevují mimo teorii grup i v umění nebo přírodě. Mnoho dekorativních vzorů používaných například ke zdobení keramiky nebo podlahových krytin využívá jednu z dihedrálních grup jako grupu symetrií. Dalším bohatým zdrojem jsou loga různých korporací (Mercedes-Benz (D_3), Chrysler (D_5)), klasická pěticípá hvězda je také příkladem grupy symetrií D_5. V přírodě můžeme ukázku symetrické grupy hledat například u ostnokožců jako jsou hvězdice nebo ježovky (známé jako mořští ježci).

[2]Inverzní symetrie:=symetrie, která po složení s původní symetrií tvoří identitu

Prvky D_n

V pravidelném n-úhelníku existuje celkem $2n$ různých symetrií: n rotací a n osových souměrností.

1. n rotací:
   pravidelný n-úhelník lze otočit o úhel $\frac{2k\pi}{n}$, kde $k \in \{0, ..., n-1\}$ tak, že se zobrazí sám na sebe.

2. n osových souměrností:
   Pro umístění os souměrností rozlišujeme případy pro n sudé a n liché:

   (a) *n je sudé*: polovina os spojuje protilehlé vrcholy a druhá polovina prochází středy protilehlých stran n-úhelníka ($n/2$ a $n/2$, celkem n os souměrnosti);

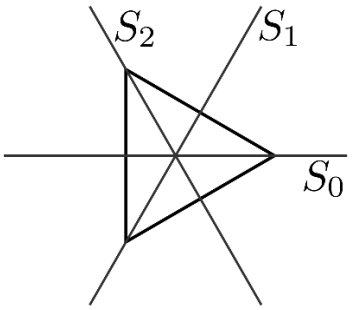
   (b) *n je liché*: osy souměrností spojují střed strany s protilehlým vrcholem(n os souměrnosti).

Grupa $D_n$obsahuje tedy minimálně $2n$ symetrií. Lze dokázat, že žádné další symetrie již neobsahuje.

Grupová operace D_n

Grupovou operací dihedrální grupy je skládání zobrazení, tedy daných symetrií. Víme, že složením dvou symetrií získáme opět symetrii. Výsledky skládání si ukážeme na příkladu rovnostranného trojúhelníka pomocí Cayleyho tabulky[3], ze které je zjevné, že dihedrální grupa nemusí být vždy komutativní. R_0 značí neutrální prvek, R_1 a R_2 rotace o 120 a 240 stupňů. S_0, S_1 a S_2 jsou osové souměrnosti vyznačené v obrázku.

[3]Cayleyho tabulka je tabulka výsledků binární operace nad konečnou množinou.



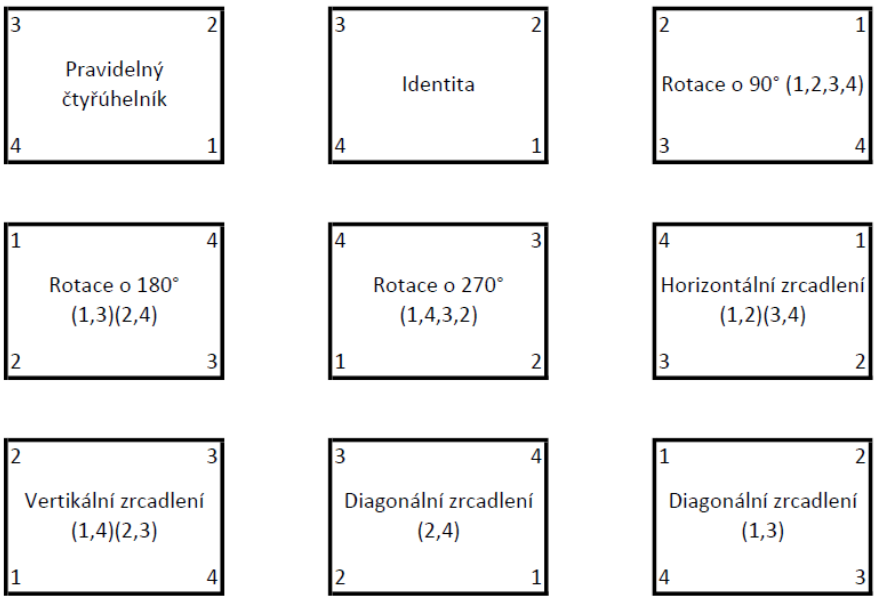
	R_0	R_1	R_2	S_0	S_1	S_2
R_0	R_0	R_1	R_2	S_0	S_1	S_2
R_1	R_1	R_2	R_0	S_1	S_2	S_0
R_2	R_2	R_0	R_1	S_2	S_0	S_1
S_0	S_0	S_2	S_1	R_0	R_2	R_1
S_1	S_1	S_0	S_2	R_1	R_0	R_2
S_2	S_2	S_1	S_0	R_2	R_1	R_0

Obrázek 3.3: Symetrie rovnostranného trojúhelníka

Na dalším příkladu si ukážeme, jak pomocí GAP nalézt prvky dané dihedrální grupy. Následně si je srovnáme s obrázkem.

```
gap> d4:= DihedralGroup(IsPermGroup,8);
Group([ (1,2,3,4), (2,4) ])

gap> Elements(d4);
[ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3), (1,3)(2,4),
 (1,4,3,2), (1,4)(2,3) ]
```



Obrázek 3.4: Symetrie v D_4

Řád D_n

Počet prvků konečné grupy G nazýváme *řád* grupy G. Označujeme jej symbolem $|G|$.

```
gap> Size(d4);
8
```

Multiplikativní tabulka D_n

Multiplikativní tabulka je využívána pro skládání dvojic symetrií dihedrální grupy D_n. Pro její určení v pravidelném n-úhelníku označíme R rotaci o úhel $\frac{2\pi}{n}$ okolo středu n-úhelníka (v kladném směru) a S osovou souměrnost podle osy procházející vrcholem (1).

Z obecného zápisu tabulky tedy vidíme, že skládání symetrií opravdu není obecně komutativní. Také vidíme, že všechny symetrie vzniklé skládáním patří mezi prvky dihedrální grupy. To znamená, že patří-li A, B do grupy D_n, patří do ní i AB. Této vlastností říkáme *uzavřenost*, tedy že grupa je *uzavřená na skládání symetrií.*

	id	R	R^2	...	$R^{(n-1)}$	S	$R°S$	$R^2°S$	...	$R^{(n-1)}°S$
id	id	R	R^2	...	$R^{(n-1)}$	S	$R°S$	$R^2°S$	...	$R^{(n-1)}°S$
R	R	R^2	$R^{3 \bmod n}$	...	id	$R°S$	$R^2°S$	$R^{3 \bmod n}°S$	...	S
R^2	R^2	$R^{3 \bmod n}$	$R^{4 \bmod n}$	...	R	$R^2°S$	$R^{3 \bmod n}°S$	$R^{4 \bmod n}°S$	...	$R°S$
...	...	...	...	...	...	...	...	...	...	...
$R^{(n-1)}$	$R^{(n-1)}$	id	R	...	$R^{(n-2)}$	$R^{(n-1)}°S$	S	$R°S$	...	$R^{(n-2)}°S$
S	S	$R^{(n-1)}°S$	$R^{(n-2)}°S$	...	$R°S$	id	$R^{(n-1)}$	$R^{(n-2)}$	...	R
$R°S$	$R°S$	S	$R^{(n-1)}°S$	...	$R^2°S$	R	id	$R^{(n-1)}$	...	$R^{2 \bmod n}$
$R^2°S$	$R^2°S$	$R°S$	S	...	$R^{3 \bmod n}°S$	R^2	R	id	...	$R^{3 \bmod n}$
...	...	...	...	...	...	...	...	...	...	...
$R^{(n-1)}°S$	$R^{(n-1)}°S$	$R^{(n-2)}°S$	$R^{(n-3)}°S$	...	S	$R^{(n-1)}$	$R^{(n-2)}$	$R^{(n-3)}$	...	id

Obrázek 3.5: Multiplikativní tabulka D_n

Postup pro skládání symetrií[4]: $(\forall i, j : 0 \leq i, j < n)$

- $R^i \circ R^j = R^{(i+j) \mod n}$
- $R^i \circ S \circ R^j = R^{(i-j) \mod n} \circ S$
- $R^i \circ R^j \circ S = R^{(i+j) \mod n} \circ S$
- $R^i \circ S \circ R^j \circ S = R^{(i-j) \mod n}$

Zdá-li se někomu postup příliš složitý nebo zdlouhavý, jistě jej potěší, že i toto zvládne GAP udělat za nás.

```
gap> ShowMultiplicationTable(DihedralGroup(IsFpGroup,8));
*     | <id>  r^-1  r     s     r^2   r*s   s*r   s*r^2
------+-----------------------------------------------
<id>  | <id>  r^-1  r     s     r^2   r*s   s*r   s*r^2
r^-1  | r^-1  r^2   <id>  s*r   r     s     s*r^2 r*s
r     | r     <id>  r^2   r*s   r^-1  s*r^2 s     s*r
s     | s     r*s   s*r   <id>  s*r^2 r^-1  r     r^2
r^2   | r^2   r     r^-1  s*r^2 <id>  s*r   r*s   s
r*s   | r*s   s*r^2 s     r     s*r   <id>  r^2   r^-1
s*r   | s*r   s     s*r^2 r^-1  r*s   r^2   <id>  r
s*r^2 | s*r^2 s*r   r*s   r^2   s     r     r^-1  <id>
```

Obrázek 3.6: Multiplikativní tabulka D_4 vytvořená softwarem GAP

[4] $x \mod n$ značí celočíselný zbytek po dělení čísla x číslem n

3.4. Příklady grup

Příklad 3.4.1. Množina celých čísel Z, množina racionálních čísel Q a množina reálných čísel R tvoří grupu společně se sčítáním. Ve všech těchto případech je neutrálním prvkem 0 a inverzním prvkem k prvku a je $-a$.

Příklad 3.4.2. Podmnožina $\{1, -1, i, -i\}$ množiny komplexních čísel C spolu s operací násobení tvoří grupu. Inverzním prvkem k prvku -1 je prvek sám, inverzní k i je $-i$

Příklad 3.4.3. Množina regulárních matic 2.stupně s operací násobení matic se nazývá *obecná lineární grupa matic 2. stupně nad množinou* R a značíme ji $GL(2, R)$.

$$GL(2, R) = \{\begin{pmatrix} a & b \\ c & d \end{pmatrix} | a, b, c, d \in R, ad - bc \neq 0\}$$

Příklad 3.4.4. Množina regulárních matic 2.stupně, jejichž determinant se rovná 1 s operací násobení matic se nazývá *speciální lineární grupa matic 2. stupně nad množinou* R a značíme ji $SL(2, R)$.

$$SL(2, R) = \{\begin{pmatrix} a & b \\ c & d \end{pmatrix} | a, b, c, d \in R, ad - bc = 1\}$$

Příklad 3.4.5. Pro pevný bod $(a, b) \in R^2$ definujeme zobrazení $T_{a,b} : R^2 \to R^2$ jako $(x, y) \to (x + a, y + b)$. Pak $G = \{T_{a,b} | a, b \in R\}$ se skládáním zobrazení je grupa. Lze ukázat, že $T_{a,b}T_{c,d} = T_{a+c,b+d}$, tedy vidíme, že G je uzavřená, neutrálním prvkem je $T_{0,0}$, inverzní prvek k $T_{a,b}$ je ne tvaru $T_{-a,-b}$ a G je Abelova. Funkce skládání je asociativní. Prvky grupy G nazveme *posunutí*.

Příklad 3.4.6. Celé číslo a má v operaci násobení inverzní prvek $\bmod n$, právě když a a n jsou nesoudělné. Pro všechna $n > 1$ definujeme množinu $U(n)$ jako množinu všech přirozených čísel menších, než n, která jsou s n nesoudělná. Potom množina $U(n)$ s operací násobení $\bmod n$ tvoří grupu. Pro $n = 10$ je množina $U(10) = \{1, 3, 7, 9\}$. Cayleyho tabulka pro $U(10)$ vypadá následovně:

*	1	3	7	9
1	1	3	7	9
3	3	9	1	7
7	7	1	9	3
9	9	7	3	1

Obrázek 3.7: Násobení v U(10) mod 10

GRUPA	OPERACE	NEUTRÁLNÍ PRVEK	TVAR PRVKU	INVERZNÍ PRVEK	ABELOVA?
Z	sčítání	0	k	(-1)k	ANO
Q^+	násobení	1	m/n; m,n>0	n/m	ANO
Z_n	sčítání mod n	0	k	n-k	ANO
R*	násobení	1	x	1/x	ANO
C*	násobení	1	a+bi	***	ANO
GL(2, M)	násobení matic	*	** ad-bc != 0	****	NE
U(n)	násobení mod n	1	k; gcd(k, n)=1	x; kx mod n = 1	ANO
R^n	posunutí	(0, 0, ..., 0)	$(a_1,a_2,...,a_n)$	$(-a_1,-a_2,...,-a_n)$	ANO
SL(2, M)	násobení matic	*	** ad-bc = 1	*****	NE
D_n	skládání	R_0	R_i, L	R_{360-i}, L	NE

*

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

**

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

***

$$\frac{1}{a^2+b^2}a - \frac{1}{a^2-b^2}bi$$

****

$$\begin{bmatrix} \frac{d}{ad-bc} & \frac{-b}{ad-bc} \\ \frac{-c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}$$

*****

$$\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Obrázek 3.8: Příklady grup (M$\in \{Q, R, C, Z_n\}$; L je osová souměrnost)

3.5. Základní vlastnosti grup

V předchozí části kapitoly jsme si ukázali různé příklady konkrétních grup. Z definice grupy plyne několik základních vlastností, ke kterým je ale nutno dodat pár poznatků.
V každé grupě existuje *právě jeden* neutrální prvek. Ke *každému* prvku grupy existuje inverzní prvek.

Věta 4. *V grupě G existuje právě jeden neutrální prvek e.*

Věta 5 (Zákony o krácení)**.** *Nechť G je grupa, $a, b, c \in G$. Pak platí*

$$ba = ca \Rightarrow b = c, ab = ac \Rightarrow b = c.$$

Věta 6. *Pro každý prvek a grupy G existuje právě jeden prvek $a' \in G$ takový, že platí $aa' = a'a = e$.*

Jistě jste si všimli, že ve zbytku textu je inverzní prvek značen jako a^{-1}. Tímto jsme se dostali k problematice *mocnin*, kde $\forall n \in \mathbb{N} : a^n = aa \cdots a$ (a se v součinu vyskytuje $n$krát), tedy

$$a^n = a \cdot a^{n-1}.$$

Speciálně $a^0 = e$.

Vlastnosti mocnin

Jestliže je n záporné celé číslo, pak $a^n = (a^{-1})^{|n|}$.
Pro všechna celá čísla m, n platí $a^m a^n = a^{m+n}$, $(a^m)^n = a^{mn}$.

Poznámka. *V nekomutativní grupě obecně $(ab)^n \neq a^n b^n$. Rovnost nastává pouze tehdy, je-li grupa Abelova.*

Věta 7. *Pro všechny prvky $a, b \in G$ platí: $(ab)^{-1} = b^{-1}a^{-1}$.*

Cvičení

1. (a) Dokažte(pomocí obrázku), že složením dvou osových souměrností vznikne rotace;

   (b) S využitím GAP složte symetrie v D_4 a zjistěte, jaké symetrie dostanete, resp. jaké rotace vzniknou složením osových souměrností;

   (c) Sestavte multiplikativní tabulku a ověřte si správnost programu.

2. (a) Zvažte, čím se stane rotace následována osovou souměrností v libovolné dihedrální grupě;

   (b) Ověřte skládání naopak, tedy pro osovou souměrnost následovanou rotací. Porovnejte výsledky skládání;

   (c) Pomocí GAP otestujte své úvahy na několika příkladech párů rotací a souměrností;

   (d) Pro lepší představu si příklad ztvárněte i pomocí náčrtu na libovolném n-úhelníku ($n \geq 4$)

3. V kapitole *Grupy* jsme si ukázali mezi příklady grup i množinu $U(n)$, která obsahuje všechna kladná celá čísla menší než n, která jsou s n nesoudělná a grupu tvoří společně s násobením $\bmod n$. K řešení následujících příkladů budete využívat funkci, která nám pro požadované n vygeneruje jí příslušnou množinu $U(n)$.

   (a) Zjistěte, kolik prvků má množina $U(n)$ pro $n \in \{9, 27, 81, 243, 5, 25, 125\}$. Rozhodněte, jaký je vztah mezi velikostmi $U(p^k)$, kde p je prvočíslo > 2 a k je kladné celé číslo;

   (b) Zjistěte, kolik prvků má množina $U(n)$ pro $n \in \{18, 54, 162, 486, 50, 250, 98, 242\}$. Rozhodněte, jaký je vztah mezi velikostmi $U(2p^k)$ a $U(p^k)$, kde p je prvočíslo > 2;

   (c) nechť r, s jsou nesoudělná čísla. Pomocí GAP určete závislost velikosti $U(rs)$ na velikostech $U(r), U(s)$.

Kapitola 4

Permutační grupy

V této kapitole budeme studovat jistou speciální skupinu grup funkcí, kterým se říká permutační grupy. V první polovině 19. století byly grupy permutací jedinými grupami vyšetřovanými tehdejšími matematiky. Teprve kolem roku 1850 byla Arthurem Cayleym představena myšlenka abstraktní grupy a trvalo dalšího čtvrt století, než se pevně ujala.

4.1. Definice a značení

Permutace množiny A je zobrazení z A do A, které je bijektivní. *Permutační grupa* na množině A je množina permutací na A, která spolu s operací skládání tvoří grupu. Ačkoliv existují grupy permutací pro různé neprázdné množiny A, my se zaměříme na případ, kdy A je konečná neprázdná množina prvků. Dále budeme předpokládat, že A je pro nějaké kladné celé číslo n množinou ve tvaru $\{1, 2, 3, \cdots n\}$. Tedy můžeme definovat jednodušeji:

Definice 3. Nechť $A = \{1, \cdots, n\}$. Bijekci $\pi : A \longrightarrow A$ říkáme *permutace* na A.

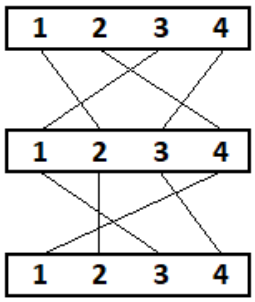
Reprezentace permutací

Permutaci $\pi(1) = 3, \pi(2) = 4, \pi(3) = 2, \pi(4) = 1$ zapisujeme jako matici

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$$

Jak jsem již na počátku kapitoly zmínila, permutace lze, jako kterákoli jiná zobrazení, skládat.

Příklad 4.1.1. $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}, \pi' = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix}$. Na následujícím obrázku je graficky znázorněno, jak složíme tyto dvě permutace.



Obrázek 4.1: Složení dvou permutací

Výsledná permutace $\pi\pi'$ je ve tvaru $\pi\pi' = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}$

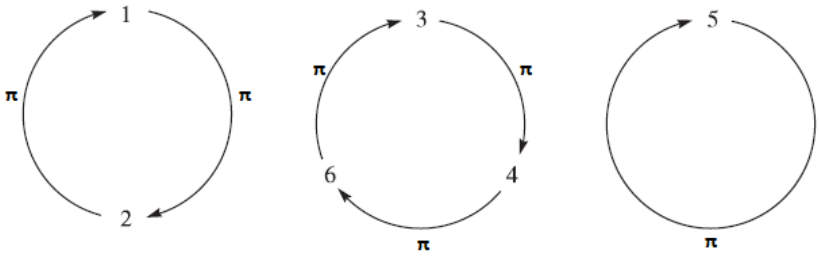
Definice 4. Nechť S_n je množina všech permutací na $A = \{1, \cdots, n\}$. Potom $S_n = (S_n, \circ)$ tvoří grupu zvanou *grupa permutací*, nebo *symetrická grupa*.

Věta 8 (Cayleyho věta)**.** *Každá konečná grupa řádu n je izomorfní s vhodnou podgrupou grupy* S_n.

4.2. Vlastnosti permutací

Nechť x_1, x_2 jsou různé prvky množiny A. *Transpozice* (x_1, x_2) je taková permutace na A, že $x_1 \mapsto x_2$ a $x_2 \mapsto x_1$ a ostatní prvky zůstávají na místě.
Dalším způsobem zápisu permutačních grup je pomocí cyklu. Tento způsob byl poprvé představen francouzským matematikem Augustinem Louisem Cauchym v roce 1815.

Způsob zápisu pomocí cyklu můžeme nejprve nastínit obrázkem.



Obrázek 4.2: Permutace pomocí cyklu

Definice 5. Nechť $x_1, \cdots, x_n$ jsou navzájem různé prvky A. *n-cyklus* $(x_1, \cdots, x_n)$ je permutace

$$(x_1, x_2) \circ (x_1, x_3) \circ \cdots \circ (x_1, x_n)$$

Příklad 4.2.1. Permutace

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 4 & 6 & 5 & 4 \end{pmatrix}$$

je součin $(1, 2) \circ (3, 4, 6) \circ (5)$

Řádem cyklu rozumíme počet jeho prvků. Je-li $\pi = \pi_1 \circ \pi_2 \circ \cdots \circ \pi_k$ rozklad na disjunktní cykly, pak řád π je nejmenší společný násobek řádů $\pi_1, \cdots \pi_k$.

Věta 9. *Každá permutace konečné množiny může být zapsána jako cyklus nebo jako produkt disjunktních cyklů.*

Kapitola 5

Konečné grupy, podgrupy

Jak si brzy ukážeme (ale z předchozího textu jsme s touto skutečností již obeznámeni), konečné grupy jsou grupy s konečným počtem prvků. Tento typ grup má velmi zajímavé vlastnosti. Nejdříve si však musíme zavést několik pojmů pro lepší porozumění následujícímu textu.

5.1. Základní pojmy a vlastnosti

Definice 6. *Řádem grupy* (konečné i nekonečné) rozumíme počet jejích prvků. Řád grupy G označujeme $|G|$.

Tedy můžeme si na příkladu ukázat, že grupa $\mathbb{Z}$ množiny celých čísel spolu se sčítáním má nekonečný řád. Naopak grupa $\mathbb{U}$ množiny $U(10) = \{1, 3, 7, 9\}$ s násobením má konečný řád a je jím číslo 4.
Řádem prvku g grupy G rozumíme nejmenší přirozené číslo n takové, že platí: $g^n = e$. V případě, že takové $n \in \mathbb{N}$ neexistuje, říkáme, že prvek má *nekonečný* řád. Řád prvku g označujeme $|g|$.
V praxi nám tedy stačí najít hodnotu exponentu, který nám v posloupnosti $g, g^2, g^3, ...$ z prvku g vytvoří neutrální prvek e. Jestliže takový exponent neexistuje, prvek g je nekonečného řádu.

Příklad 5.1.1. Ukažme si hledání řádu grupy a prvku dané grupy na příkladu s grupou $\mathbb{U}$ množiny $U(15)$, jejíž prvky jsou $\{1, 2, 4, 7, 8, 11, 13, 14\}$ s operací násobení modulo 15. Grupa G má řád 8. Nyní chceme najít řád prvku 7, tedy postupně počítáme hodnoty $7^n, n \in \mathbb{N}$:
$7^1 = 7 \equiv 7(mod15), 7^2 = 49 \equiv 4(mod15), 7^3 = 343 \equiv 13(mod15), 7^4 = 2401 \equiv 1(mod15)$. Zjistili jsme tedy, že řádem prvku 7 je 4, $|7| = 4$.

V předchozí kapitole, kde jsme si ukazovali příklady grup jste si mohli všimnout, že některé grupy jsou definovány na množině, která je podmnožinou jiné množiny, která tvoří grupu. Například *speciální lineární grupa* a *obecná lineární grupa*, kdy s jistotou můžeme říct, že množina $SL(2, \mathbb{R})$ je podmnožinou množiny $GL(2, \mathbb{R})$. Zobecněme si tento příklad zavedením pojmu *podgrupa*.
Tvoří-li podmnožina H grupy G spolu se zúženou operací[1] grupy G samostatnou grupu, říkáme, že H je *podgrupou* grupy G. Skutečnost, že H je podgrupou G označujeme $H \trianglelefteq G$.
Ověřit, zda je daná podmnožina H množiny G podgrupou G můžeme v několika snadných krocích, ve kterých využíváme vlastnosti podgrup.

Věta 10. *Nechť G je grupa a H je neprázdnou podmnožinou množiny G. Jestliže $\forall a, b \in H$ platí $ab^{-1} \in H$, pak je H podgrupou v G. V případě aditivního zápisu platí, že H je podgrupou G právě tehdy, když $\forall a, b \in H : a - b \in H$.*

Příklad 5.1.2. Mějme Abelovu grupu $G = (G, \cdot)$ s neutrálním prvkem e. Pak podmnožina H, pro kterou platí $H = \{x^2 | x \in G\}$ je podgrupou grupy G právě tehdy, když neutrální prvek e leží v H a pro všechny dvojice prvků a, b platí, že $a^2(b^2)^{-1} \in H$. Víme, že $e^2 = e$, tedy první podmínka je splněna. Nyní musíme dokázat, že $a^2(b^2)^{-1}$ je druhou mocninou nějakého prvku. Víme, že G je Abelova, můžeme tedy zapsat $a^2(b^2)^{-1}$ jako $(ab^{-1})^2$, což je regulérní zápis prvku ležícího v množině H. Tedy jsme dokázali, že H je podgrupou G.

Věta 11. *Nechť G je grupa a H je neprázdná podmnožina množiny G. Jestliže $\forall a, b \in H$ platí $ab \in H, a^{-1} \in H$, H je podgrupou grupy G.*

[1] restrikce zobrazení:=zobrazení, které má menší definiční obor, než původní zobrazení

Příklad 5.1.3. Nechť $G = (G, \cdot)$ je grupa, kde G je množinou reálných čísel bez nuly s operací násobení. Dále mějme podmnožiny H, K, $H = \{x \in G | x = 1$ nebo x je iracionální$\}$, $K = \{x \in G | x \geq 1\}$. Pak H není podgrupou G, protože $\sqrt{2} \in H$, ale $\sqrt{2} \cdot \sqrt{2} \notin H$. Také o K musíme říci, že není podgrupou v G. $2 \in K$, ale $2^{-1} \notin K$.

Věta 12. *Nechť H je neprázdná konečná podmnožina grupy G. Je-li H uzavřena na operaci grupy G, pak je podgrupou v G.*

Definice 7. Nechť H je podmnožinou grupy G. Symbolem $\langle H \rangle$ označíme průnik všech podgrup grupy G obsahujících množinu H. $\langle H \rangle$ je nejmenší podgrupou grupy G obsahující množinu H. Nazýváme ji *podgrupa generovaná množinou H.* Množinu H pak nazýváme *množinou generátorů grupy* $\langle H \rangle$.

Věta 13 (Lagrangeova věta)**.** *Nechť H je podgrupou konečné grupy G. Potom platí:*

$$|G| = [G : H] \cdot |H|.$$

Z Lagrangeovy věty plyne, že řád podgrupy H dělí řád grupy G.
Jednou z nejdůležitějších podgrup je *centrum grupy G*, které označíme jako $Z(G)$. Jedná se o podgrupu v G, jejíž každý prvek komutuje s každým prvkem v G. Tedy symbolicky

$$Z(G) = \{a \in G | \forall x \in G : ax = xa\}.$$

Příklad 5.1.4. Pro $n \geq 3$,

$$Z(D_n) = \begin{cases} \{R_0, R_{180}\} & \text{pokud } n \text{ je sudé} \\ \{R_0\} & \text{pokud } n \text{ je liché} \end{cases}$$

1. Vyšetřete vlastnosti dihedrální grupy D_4;

   Řešení:

```
gap> G:=DihedralGroup(IsPermGroup, 16);
Group([ (1,2,3,4,5,6,7,8), (2,8)(3,7)(4,6) ])
gap> Elements(G);
[ (), (2,8)(3,7)(4,6),(1,2)(3,8)(4,7)(5,6),
 (1,2,3,4,5,6,7,8),(1,3)(4,8)(5,7),(1,3,5,7)
 (2,4,6,8),(1,4)(2,3)(5,8)(6,7),(1,4,7,2,5,8,3,6),
 (1,5)(2,4)(6,8),(1,5)(2,6)(3,7)(4,8),(1,6)
 (2,5)(3,4)(7,8),(1,6,3,8,5,2,7,4),(1,7)(2,6)(3,5),
 (1,7,5,3)(2,8,6,4),(1,8,7,6,5,4,3,2),(1,8)(2,7)
 (3,6)(4,5) ]
gap> a:=G.1;
(1,2,3,4,5,6,7,8)
gap> b:=G.2;
(2,8)(3,7)(4,6)
gap> H:=Subgroup(G, [a]);
Group([ (1,2,3,4,5,6,7,8) ])
gap> Elements(H);
[ (), (1,2,3,4,5,6,7,8), (1,3,5,7)(2,4,6,8),
 (1,4,7,2,5,8,3,6), (1,5)(2,6)(3,7)(4,8),
 (1,6,3,8,5,2,7,4),(1,7,5,3)(2,8,6,4),
 (1,8,7,6,5,4,3,2) ]
gap>
```

   - V prvním řádku programu přiřazujeme identifikátoru G jako hodnotu dihedrální grupu řádu 16. Z následujícího řádku - odpovědi programu - vidíme, že prvky $(1,2,3,4,5,6,7,8), (2,8)(3,7)(4,6)$ jsou generátory dané grupy.
   - V dalším příkazu jsme po GAP chtěli vypsání prvků grupy G na obrazovku. Vidíme, že oba generátory jsou obsaženy mezi prvky.
   - Na dalších řádcích přiřazujeme identifikátorům a, b generátory naší grupy G a do hodnoty H navazujeme podgrupu grupy G generovanou pomocí dříve zavedeného a.
   - Poslední příkaz nám na obrazovku vypsal prvky podgrupy H.

```
gap> K:= Subgroup(G, [a, b]);
Group([ (1,2,3,4,5,6,7,8), (2,8)(3,7)(4,6) ])
gap> Size(K);
16
gap> Elements(K);
[ (),(2,8)(3,7)(4,6),(1,2)(3,8)(4,7)(5,6),
 (1,2,3,4,5,6,7,8),(1,3)(4,8)(5,7),(1,3,5,7)
 (2,4,6,8),(1,4)(2,3)(5,8)(6,7),(1,4,7,2,5,8,3,6),
 (1,5)(2,4)(6,8),(1,5)(2,6)(3,7)(4,8),
 (1,6)(2,5)(3,4)(7,8),(1,6,3,8,5,2,7,4),
 (1,7)(2,6)(3,5),(1,7,5,3)(2,8,6,4),
 (1,8,7,6,5,4,3,2),(1,8)(2,7)(3,6)(4,5) ]
gap>
```

- Na prvním řádku uvedeného programu jsme identifikátoru K přiřadili podgrupu G generovanou prvky a, b. Z příkazu Elements(K), Size(K) vidíme, že $K = G$.
- Následující příkazy je dobré znát při práci s GAP:
  (a) chceme-li najít centrum grupy G:$Center(G)$;
  (b) chceme-li najít řád prvku a grupy G: $Order(a)$;
  (c) chceme-li zjistit, zda je grupa G Abelova: $IsAbelian(G)$;
  (d) chceme-li zjistit, zda je grupa G cyklická: $IsCyclic(G)$.

Cvičení

2. S pomocí GAP určete řád prvků $3, 7, 13$ a 97 v $U(100)$. Určete také řád prvků k nim inverzních;
3. S pomocí GAP určete řád prvků $3, 13, 153$ a 317 v $U(430)$. Určete také řád prvků k nim inverzních;
4. Určete, jaký je vztah mezi řádem prvku a řádem prvku k němu inverzním.

Kapitola 6

Cyklické grupy

S vědomostmi z předchozí kapitoly můžeme definovat grupu cyklickou. Grupa G se nazývá *cyklická*, pokud v ní existuje prvek a takový, že $G = \{a^n | n \in \mathbb{Z}\}$. Daný prvek a se nazývá *generátor* grupy G. Díky zavedené notaci můžeme zapsat $G = \langle a \rangle$ a číst, že cyklická grupa G je generovaná prvkem a.

6.1. Vlastnosti cyklických grup

V této části kapitoly shrneme základní vlastnosti cyklických grup. Jak taková cyklická grupa vypadá si můžeme ukázat na příkladu.

Příklad 6.1.1. Množina celých čísel $\mathbb{Z}$ s operací sčítání tvoří cyklickou grupu generovanou prvky $\{-1, 1\}$ a platí, že $1^n = 1 + 1 + \cdots + 1$, pro n kladné a $1^n = (-1) + (-1) + \cdots + (-1)$ pro n záporné.

Příklad 6.1.2. $\mathbb{Z}_8 = \langle 1 \rangle = \langle 3 \rangle = \langle 5 \rangle = \langle 7 \rangle =$. Skutečnost ověříme pro $\mathbb{Z}_8 = \langle 3 \rangle$, kde $\langle 3 \rangle = \{3, 3+3, 3+3+3, \cdots\}$ určuje množinu $\{3, 6, 1, 4, 7, 2, 5, 0\}$, která se rovná $\mathbb{Z}_8$.
Naopak prvek 2 není generátorem grupy $\mathbb{Z}_8$, protože $\langle 2 \rangle = \{0, 2, 4, 6\} \neq \mathbb{Z}_8$.

Věta 14. *Nechť G je grupa, $a \in G$. Jestliže má prvek a nekonečný řád, pak $a^i = a^j$ tehdy a jen tehdy, když $i = j$. Pokud má a konečný řád (ozn. n), pak $\langle a \rangle = \{e, a, a^2, \cdots, a^{n-1}\}$ a $a^i = a^j$ tehdy a jen tehdy, když n dělí $i - j$.*

Zde si uvedeme několik důsledků spojených s řády prvků cyklické grupy:

- Pro každý prvek a grupy platí, že $|a| = |\langle a \rangle|$;
- Nechť G je grupa, $a \in G, |a| = n$. Pokud $a^k = e$, pak n dělí k.

Věta 15. *Nechť a je prvek řádu n v grupě G a nechť k je kladné celé číslo. Pak platí $\langle a^k \rangle = \langle a^{gcd(n,k)} \rangle$ a $|a^k| = \frac{n}{gcd(n,k)}$*

Výhodou vyslovené věty je, že nám umožňuje nahradit jeden generátor cyklické podgrupy jiným. Například pro $|a| = 30$ máme $\langle a^{26} \rangle = \langle a^2 \rangle, \langle a^{23} \rangle = \langle a \rangle, \langle a^{21} \rangle = \langle a^3 \rangle$. Opět nám vyplývají některé důsledky týkající se spojitosti mezi řádem konečné cyklické grupy a řádem jejího prvku:

- Nechť $|a| = n$. Pak $\langle a^i \rangle = \langle a^j \rangle$ právě tehdy, když $gcd(n, i) = gcd(n, j)$ a současně platí, že $|a^i| = |a^j|$ právě tehdy, když $gcd(n, i) = gcd(n, j)$.
- Nechť $|a| = n$. Pak $\langle a \rangle = \langle a^j \rangle$ právě tehdy, když $gcd(n, j) = 1$ a $|a| = |\langle a^j \rangle|$ právě tehdy, když $gcd(n, j) = 1$.
- Číslo k je generátorem v $\mathbb{Z}_n$ tehdy a jen tehdy, když $gcd(n, k) = 1$.

1. Vyšetřete vlastnosti cyklické grupy C_6

   Řešení:

   Cyklickou grupu řádu n si můžeme jednoduše představit jako grupu všech mocnin n-cyklu $(1, 2, ..., n)$. Na následujícím příkladu si ukážeme, jak GAP sestaví cyklickou grupu pro $n = 6$:

```
gap> c6:=CyclicGroup(IsPermGroup, 6);
Group([ (1,2,3,4,5,6) ])
gap> Elements(c6);
[ (), (1,2,3,4,5,6), (1,3,5)(2,4,6), (1,4)(2,5)(3,6),
 (1,5,3)(2,6,4), (1,6,5,4,3,2) ]
gap> a:=c6.1;
(1,2,3,4,5,6)
gap> Elements(Subgroup(c6, [a^2]));
[ (), (1,3,5)(2,4,6), (1,5,3)(2,6,4) ]
gap>
```

   - První dva řádky již není třeba komentovat. Na třetím řádku jsme opět určitému identifikátoru (a) přiřadili jako hodnotu jeden z prvků grupy $c6$.
   - Na čtvrtém řádku je GAP předán příkaz pro vypsání prvků podgrupy grupy $c6$ generované prvkem a^2.
   - Z předchozí kapitoly víme, že GAP zvládne sestavit podgrupu z více prvků, než z jednoho:

```
gap> Elements(Subgroup(c6, [a^2, a^3]));
[ (), (1,2,3,4,5,6),(1,3,5)(2,4,6),
 (1,4)(2,5)(3,6),(1,5,3)(2,6,4),(1,6,5,4,3,2) ]
gap>
```

   - Vidíme, že podgrupa generovaná prvky a^2, a^3 je rovna grupě $c6$.

Cvičení

2. S pomocí GAP najděte všechny podgrupy následujících grup:

   (a) D_4

   (b) cyklická podgrupa grupy D_8 generovaná $(1, 2, 3, 4, 5, 6, 7, 8)$

3. Napište funkci, která bere jako vstup cyklickou grupu a jako výstup vrací četnost jednotlivých řádů prvků v grupě.

   (a) Najděte počet prvků jednotlivých řádů v cyklických grupách řádu 75 a 90;

   (b) Najděte počet prvků jednotlivých řádů v dihedrálních grupách D_{17}, D_{25}, D_{33} a D_{49}. Rozhodněte, zda platí nějaký vztah pro počet prvků řádu 2 v grupě D_n;

   (c) Najděte počet prvků řádu 2 v dihedrálních grupách D_{18}, D_{26}, D_{34} a D_{50}. Rozhodněte, zda platí nějaký vztah pro počet prvků řádu 2 v grupě D_n;

4. Rozhodněte, jaký vztah platí mezi řádem grupy a řády jejích prvků.

Řešení úloh

Úvod do programu GAP

1. Základní funkce GAP

   Poznámka. *Pro výpočet jednoduchých hodnot výrazů či jejich porovnání není třeba v GAP definovat funkce, požadovaný výraz stačí programem pouze vyhodnotit. Při srovnávání hodnot vrací GAP pravdivostní hodnotu.*

   (a) Vypočtěte hodnotu 3^{121}

   (b) Zjistěte, zda $2^{25} + (45 * 51777)$ je větší, než 34 milionů;

```
gap> 3^121;
53910308997432936312395394885288151191944268826135533192 03

gap> ((2^25) + (45*51777)) > 34*10^6;
true
```

2. Příkaz přiřazení v GAP

   Poznámka. *Operace s identifikátory probíhají stejně, jako v předchozím cvičení, jen místo konkrétních hodnot používáme navázané symboly. Toto je při složitějších či zdlouhavějších výpočtech velmi praktické.*

   (a) Přiřaďte hodnotu 14 identifikátoru b

   (b) Přiřaďte hodnotu 123456789 identifikátoru $BigNUMBER$

   (c) Vypočtěte $b + BigNUMBER$

```
gap> b:=14;;
gap> BigNUMBER:=123456789;;
gap> b+BigNUMBER;
123456803
```

Úvodní pojmy

1. Vlastnosti celých čísel

   Poznámka. *K výpočtu Nsd (Gcd) a nsn (Lcm) opět využíváme pouze předdefinované funkce, které jsme si ukázali v úvodním manuálu programu.*

   (a) Vypočítejte největšího společného dělitele a nejmenší společný násobek:

   i. čísel 25646 a 68185

   ii. čísel 9245 a 325

```
gap> Gcd(25646, 68185);
1
gap> Lcm(25646, 68185);
1748672510
gap> Gcd(9245, 325);
5
gap> Lcm(9245, 325);
600925
```

   Poznámka. *Jak jsem již v textu uvedla, funkce* $Gcdex(x, y)$ *nám vrací celkem čtyři koeficienty, z nichž:*

   i. $coeff1 = s$, $s \cdot x + t \cdot y = Gcd(x, y)$;

   ii. $coeff2 = t$, $s \cdot x + t \cdot y = Gcd(x, y)$;

   iii. $coeff3 = m$, $m \cdot x + n \cdot y = 0$;

   iv. $coeff4 = n$, $m \cdot x + n \cdot y = 0$;

(b) Použijte funkci *Gcdex* k nalezení koeficientů s, t pro které platí, že $gcd(8701, 10057) = 8701s + 10057t$

```
gap> Gcdex(8701, 10057);
rec( coeff1 := 37, coeff2 := -32, coeff3 := -89,
 coeff4 := 77, gcd := 113 )
```

Hledané koeficienty jsou s=37, t=-32;

2. Funkce

(a) Vytvořte funkci, která přijímá jako argument kladné celé číslo n a vrací hodnotu součtu $1 + 2 + \cdots + n$. Poté využijte tuto funkci pro $n = 6$ a $n = 2584$.

Poznámka. *K řešení využijeme vzorec pro součet aritmetické posloupnosti:* $s_n = \frac{n}{2}(a_1 + a_n)$

```
 gap> newfunction:=function(n)
> local x;
> x:=(n*(n+1))/2;
> return x;
> end;
function( n ) ... end
gap> newfunction(6);
21
gap> newfunction(2584);
3339820
```

(a) Vytvořte funkci, která přijímá jako argument libovolné celé číslo t a vrací hodnotu

$$f(t) = t^3 + 3 - 2t$$

;

Poznámka. *Opět k vyřešení nepotřebujeme nic jiného, než daný předpis pro funkci. Výraz jednoduše zapíšeme do hodnoty proměnné ve funkci a ta si jej sama vypočítá ihned po tom, co bude zavolána s nějakou konkrétní hodnotou t.*

```
gap> newfunction2:=function(t)
> local x;
> x:=(t^3)+3-(2*t);
> return x;
> end;
function( t ) ... end
gap> newfunction2(6);
207
gap> newfunction2(2584);
17253507539
```

(a) Vytvořte funkci, která ověří validitu zadaného rodného čísla;

   i. použijte Vámi vytvořenou funkci pro zkontrolování Vašeho rodného čísla;

   ii. změňte v zadaném čísle jednu číslici. Dokázala Vámi napsaná funkce rozeznat chybné zadání?

Poznámka. *Příklad řešíme s využitím znalostí modulární aritmetiky. Hodnotu rodného čísla modulo* 11 *položíme rovnu nule a tím pádem nám funkce vrací pravdivostní hodnotu, zda je vložené číslo validní, či nikoliv;*

```
gap> RCvalidity:=function(r)
> local x;
> x:= r mod 11 = 0;
> return x;
> end;
function( r ) ... end
gap> RCvalidity(9555315649);
true
gap> RCvalidity(9555305649);
false
```

Úvod do teorie grup

1. (a) S využitím GAP složte symetrie v D_4 a zjistěte, jaké symetrie dostanete, resp. jaké rotace vzniknou složením osových souměrností;

   (b) Ověřte skládání naopak, tedy pro osovou souměrnost následovanou rotací. Porovnejte výsledky skládání;

   (c) Pomocí GAP otestujte své úvahy na několika příkladech párů rotací a souměrností;

   (d) Sestavte multiplikativní tabulku a ověřte si správnost programu.

   Poznámka. *Pro přehlednější a jednodušší manipulaci si symetrie ve formě závorek (permutací) můžeme navázat na identifikátory a operovat s nimi; V druhé části příkladu jsem symetrie ponechala ve formě permutací a sami vidíte, že práce s identifikátory je opravdu jednodušší.*

```
gap> d4:= DihedralGroup(IsPermGroup,8);
Group([ (1,2,3,4), (2,4) ])
gap> Elements(d4);
[ (), (2,4), (1,2)(3,4), (1,2,3,4), (1,3), (1,3)(2,4),
  (1,4,3,2), (1,4)(2,3) ]
```

```
gap> o1:=(1,2)(3,4);;
gap> o2:=(1,4)(2,3);;
gap> o1*o2;
(1,3)(2,4)
gap> ()*(2,4);
(2,4)
gap> (2,4)*(1,2)(3,4);
(1,2,3,4)
gap> (1,2,3,4)*(1,4,3,2);
()
gap> (1,4)(2,3)*(1,3);
(1,4,3,2)
```

Poznámka. *V GAPem vygenerované multiplikativní tabulce vidíme, jak by program dopadl pro ostatní symetrie, není tedy třeba je zde v řešení rozepisovat.*

```
gap> ShowMultiplicationTable(DihedralGroup(IsPermGroup,8));
*          | ()         (2,4)      (1,2)(3,4) (1,2,3,4)  (1,3)      (1,3)(2,4) (1,4,3,2)  (1,4)(2,3)
-----------+-----------------------------------------------------------------------------------------
()         | ()         (2,4)      (1,2)(3,4) (1,2,3,4)  (1,3)      (1,3)(2,4) (1,4,3,2)  (1,4)(2,3)
(2,4)      | (2,4)      ()         (1,2,3,4)  (1,2)(3,4) (1,3)(2,4) (1,3)      (1,4)(2,3) (1,4,3,2)
(1,2)(3,4) | (1,2)(3,4) (1,4,3,2)  ()         (1,3)      (1,2,3,4)  (1,4)(2,3) (2,4)      (1,3)(2,4)
(1,2,3,4)  | (1,2,3,4)  (1,4)(2,3) (2,4)      (1,3)(2,4) (1,2)(3,4) (1,4,3,2)  ()         (1,3)
(1,3)      | (1,3)      (1,3)(2,4) (1,4,3,2)  (1,4)(2,3) ()         (2,4)      (1,2)(3,4) (1,2,3,4)
(1,3)(2,4) | (1,3)(2,4) (1,3)      (1,4)(2,3) (1,4,3,2)  (2,4)      ()         (1,2,3,4)  (1,2)(3,4)
(1,4,3,2)  | (1,4,3,2)  (1,2)(3,4) (1,3)      ()         (1,4)(2,3) (1,2,3,4)  (1,3)(2,4) (2,4)
(1,4)(2,3) | (1,4)(2,3) (1,2,3,4)  (1,3)(2,4) (2,4)      (1,4,3,2)  (1,2)(3,4) (1,3)      ()
```

2. (a) Zjistěte, kolik prvků má množina $U(n)$ pro $n \in \{9, 27, 81, 243, 5, 25, 125\}$. Rozhodněte, jaký je vztah mezi velikostmi $U(p^k)$, kde p je prvočíslo > 2 a k je kladné celé číslo;

   (b) Zjistěte, kolik prvků má množina $U(n)$ pro $n \in \{18, 54, 162, 486, 50, 250, 98, 242\}$. Rozhodněte, jaký je vztah mezi velikostmi $U(2p^k)$ a $U(p^k)$, kde p je prvočíslo > 2;

   (c) nechť r, s jsou nesoudělná čísla. Pomocí GAP určete závislost velikosti $U(rs)$ na velikostech $U(r), U(s)$.

Poznámka. *Vytváříme funkci, která pro vstupní hodnotu (n) hledá a filtruje všechna přirozená čísla, která jsou menší než n a platí, že největší společný dělitel daného čísla a n je roven jedné, tedy, že čísla jsou nesoudělná.*
Poté již stačí využít předdefinovanou funkci Size.

```
gap> ulist:=function(n)
> local s,i,o;
> o:=One(Integers mod n);
> s:=n->Filtered([1..n-1], i->Gcd(i,n)=1);
> return s(n)*o;
> end;
function( n ) ... end
gap> ulist(60);
[ ZmodnZObj( 1, 60 ),ZmodnZObj( 7, 60 ),ZmodnZObj( 11, 60 ),
  ZmodnZObj( 13, 60 ),ZmodnZObj( 17, 60 ),ZmodnZObj( 19, 60 ),
  ZmodnZObj( 23, 60 ),ZmodnZObj( 29, 60 ),ZmodnZObj( 31, 60 ),
  ZmodnZObj( 37, 60 ),ZmodnZObj( 41, 60 ),ZmodnZObj( 43, 60 ),
  ZmodnZObj( 47, 60 ),ZmodnZObj( 49, 60 ),ZmodnZObj( 53, 60 ),
  ZmodnZObj( 59, 60 ) ]
gap> Size(ulist(5));
4
gap> Size(ulist(25));
20
gap> Size(ulist(125));
100
```

Poznámka. *Ze získaných hodnot odvodíme:*

- $U(n) = m$
- $U(n^2) = n * m$
- $U(n^3) = n * n * m$
- $U(n^4) = n * n * n * m$
- $\rightarrow U(n^k) = m * n^(k-1)$

```
gap> Size(ulist(18));
6
gap> Size(ulist(54));
18
gap> Size(ulist(162));
54
```

Poznámka. *Odvodíme: U(2n)=U(n)*

```
gap> Gcd(17, 26);
1
gap> Size(ulist(17));
16
gap> Size(ulist(26));
12
gap> Size(ulist(17*26));
192
```

Poznámka. *Ze získaných hodnot odvodíme:*

- $U(r) = m$
- $U(s) = n$
- $U(r * s) = m * n$

Konečné grupy, podgrupy

1. S pomocí GAP určete řád prvků $3, 7, 13$ a 97 v $U(100)$. Určete také řád prvků k nim inverzních;

2. Určete, jaký je vztah mezi řádem prvku a řádem prvku k němu inverzním.

Poznámka. *Využíváme již dříve definovanou funkci ulist a posléze předdefinovanou funkci GAP Order a Inverse.*
Čtenář si musí dát pozor na to, že GAP vnímá jako určující hodnotu prvku jeho pořadí mezi prvky. Tedy pro nalezení řádu čísla 7 *nehledáme* $Order(e[7])$*, ale* $Order(e[3])$*.*

```
gap> e:=ulist(100);
[ ZmodnZObj( 1, 100 ), ZmodnZObj( 3, 100 ),
  ZmodnZObj( 7, 100 ), ZmodnZObj( 9, 100 ),
  ZmodnZObj( 11, 100 ), ZmodnZObj( 13, 100 ),
  ZmodnZObj( 17, 100 ), ZmodnZObj( 19, 100 ),
  ZmodnZObj( 21, 100 ), ZmodnZObj( 23, 100 ),
  ZmodnZObj( 27, 100 ), ZmodnZObj( 29, 100 ),
  ZmodnZObj( 31, 100 ), ZmodnZObj( 33, 100 ),
  ZmodnZObj( 37, 100 ), ZmodnZObj( 39, 100 ),
  ZmodnZObj( 41, 100 ), ZmodnZObj( 43, 100 ),
  ZmodnZObj( 47, 100 ), ZmodnZObj( 49, 100 ),
  ZmodnZObj( 51, 100 ), ZmodnZObj( 53, 100 ),
  ZmodnZObj( 57, 100 ), ZmodnZObj( 59, 100 ),
  ZmodnZObj( 61, 100 ), ZmodnZObj( 63, 100 ),
  ZmodnZObj( 67, 100 ), ZmodnZObj( 69, 100 ),
  ZmodnZObj( 71, 100 ), ZmodnZObj( 73, 100 ),
  ZmodnZObj( 77, 100 ), ZmodnZObj( 79, 100 ),
  ZmodnZObj( 81, 100 ), ZmodnZObj( 83, 100 ),
  ZmodnZObj( 87, 100 ), ZmodnZObj( 89, 100 ),
  ZmodnZObj( 91, 100 ), ZmodnZObj( 93, 100 ),
  ZmodnZObj( 97, 100 ), ZmodnZObj( 99, 100 ) ]
```

```
gap> e[2];
ZmodnZObj( 3, 100 )
gap> e[3];
ZmodnZObj( 7, 100 )
gap> e[6];
ZmodnZObj( 13, 100 )
gap> e[39];
ZmodnZObj( 97, 100 )

gap> Order(e[2]);
20
gap> Order(e[3]);
4
gap> Order(e[6]);
20
gap> Order(e[39]);
20
gap> Inverse(e[2]);
ZmodnZObj( 67, 100 )
gap> Inverse(e[3]);
ZmodnZObj( 43, 100 )
gap> Inverse(e[6]);
ZmodnZObj( 77, 100 )
gap> Inverse(e[39]);
ZmodnZObj( 33, 100 )
gap> e[27];
ZmodnZObj( 67, 100 )
gap> Order(e[27]);
20
gap> e[18];
ZmodnZObj( 43, 100 )
gap> Order(e[18]);
4
```

Poznámka. *Odvodíme: Řád prvku se rovná řádu prvku k němu inverznímu.*

Cyklické grupy

1. S pomocí GAP najděte všechny podgrupy následujících grup:

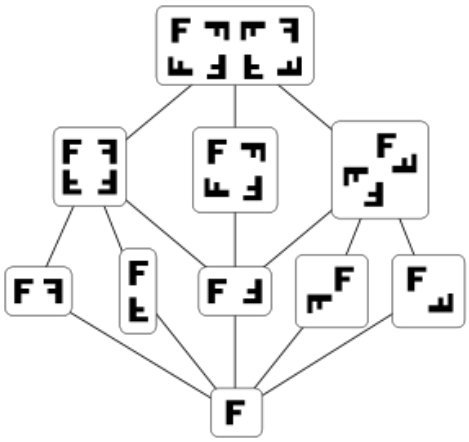
   (a) D_4

   (b) cyklická podgrupa grupy D_8 generovaná $(1, 2, 3, 4, 5, 6, 7, 8)$

   Poznámka. *Na řešení příkladu je ukázáno jednak jak získat podgrupu generovanou prvkem grupy, tak i jak si díky GAP ukázat její prvky; Pomocí d4.1, d4.2 přistupujeme k prvnímu a druhému generátoru grupy.*

```
gap> d4:=DihedralGroup(IsPermGroup,8);
Group([ (1,2,3,4), (2,4) ])
gap> a:=d4.1;
(1,2,3,4)
gap> b:=d4.2;
(2,4)
gap> Subgroup(d4,[a]);
Group([ (1,2,3,4) ])
gap> Elements(Subgroup(d4,[b]));
[ (), (2,4) ]
```

   Poznámka. *Podgrupy v grupě je možné si znázornit pomocí grafu;*
   Ve vrcholech nalezneme tzv. triviální podgrupy, tedy ty podgrupy, které obsahují buď všechny prvky grupy, nebo pouze neutrální prvek - identitu;
   Dvojice vrcholů, které jsou v tomto grafu spojeny hranou, představují grupu a její podgrupu;
   Takovýto graf nám v paměti programu vytvoří příkaz LatticeSubgroups.



Obrázek 6.1: Podgrupy dihedrální grupy D4

Poznámka. *Na obrázku jsou symetrie v D_4 znázorněny pomocí transformací písmene F;*

Zdroj: ***https://cs.wikipedia.org/wiki/Grupa***

```
gap> d8SUB:=Subgroup(d8, [(1,2,3,4,5,6,7,8)]);
Group([ (1,2,3,4,5,6,7,8) ])
gap> L:=LatticeSubgroups(d8SUB);
<subgroup lattice of Group([ (1,2,3,4,5,6,7,8) ]),
 4 classes, 4 subgroups>
gap> ConjugacyClassesSubgroups(L);
[ Group( [ () ] )^G, Group( [ (1,5)(2,6)
 (3,7)(4,8) ] )^G,
  Group( [ (1,3,5,7)(2,4,6,8) ] )^G,
  Group( [ (1,2,3,4,5,6,7,8) ] )^G ]
```

2. Napište funkci, která bere jako vstup cyklickou grupu a jako výstup vrací četnost jednotlivých řádů prvků v grupě.

Poznámka. *Opět tvoříme funkci pomocí funkcí, které jsou již v GAP předdefinovány - Elements, Order; Stejně jako v předchozí funkci nám program vrací seznam uspořádaných dvojic ve tvaru Řád prvku : Počet prvků daného řádu.*

```
gap> orderFreq:=function(g)
> local h,w;
> w:=[];
> w:=h->Collected(List(Elements(h),Order));
> Print("[Order␣of␣element,␣Number␣of␣that␣order]=");
> return w(g);
> end;
function( g ) ... end
```

(a) Najděte počet prvků jednotlivých řádů v cyklických grupách řádu 75 a 90;

```
gap> c75:=CyclicGroup(IsPermGroup,75);
Group([
 (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
  20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,
  36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,
  52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,
  68,69,70,71,72,73,74,75) ])
gap> orderFreq(c75);
[Order of element, Number of that order]=[
 [ 1, 1 ], [ 3, 2 ], [ 5, 4 ], [ 15, 8 ],
 [ 25, 20 ], [ 75, 40 ] ]

gap> c90:=CyclicGroup(IsPermGroup,90);
Group([
 (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
  20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,
  36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,
  52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,
  68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,
  84,85,86,87,88,89,90) ])

gap> orderFreq(c90);
[Order of element, Number of that order]=[ [ 1, 1 ],
 [ 2, 1 ], [ 3, 2 ], [ 5, 4 ], [ 6, 2 ], [ 9, 6 ],
 [ 10, 4 ], [ 15, 8 ], [ 18, 6 ], [ 30, 8 ],
 [ 45, 24 ], [ 90, 24 ] ]
```

(b) Najděte počet prvků jednotlivých řádů v dihedrálních grupách D_{17}, D_{25}, D_{33} a D_{49}. Rozhodněte, zda platí nějaký vztah pro počet prvků řádu 2 v grupě D_n;

```
gap> d17:=DihedralGroup(IsPermGroup,34);;
gap> d25:=DihedralGroup(IsPermGroup,50);;
gap> d33:=DihedralGroup(IsPermGroup,66);;
gap> orderFreq(d17);
[Order_of_element,_Number_of_that_order]=[ [ 1, 1 ],
 [ 2, 17 ], [ 17, 16 ] ]
gap> orderFreq(d25);
[Order_of_element,_Number_of_that_order]=[ [ 1, 1 ],
 [ 2, 25 ], [ 5, 4 ], [ 25, 20 ] ]
gap> orderFreq(d33);
[Order_of_element,_Number_of_that_order]=[ [ 1, 1 ],
 [ 2, 33 ], [ 3, 2 ], [ 11, 10 ], [ 33, 20 ] ]
```

Poznámka. *Pro dihedrální grupy D_n platí, že počet prvků řádu 2 je roven n;*

(c) Najděte počet prvků řádu 2 v dihedrálních grupách D_{18}, D_{26}, D_{34} a D_{50}. Rozhodněte, zda platí nějaký vztah pro počet prvků řádu 2 v grupě D_{2n};

3. Rozhodněte, jaký vztah platí mezi řádem grupy a řády jejích prvků.

```
gap> d34:=DihedralGroup(IsPermGroup,68);;
gap> d26:=DihedralGroup(IsPermGroup,52);;
gap> d18:=DihedralGroup(IsPermGroup,36);;
gap> orderFreq(d34);
[Order_of_element,_Number_of_that_order]=[ [ 1, 1 ],
 [ 2, 35 ], [ 17, 16 ], [ 34, 16 ] ]
gap> orderFreq(d26);
[Order_of_element,_Number_of_that_order]=[ [ 1, 1 ],
 [ 2, 27 ], [ 13, 12 ], [ 26, 12 ] ]
gap> orderFreq(d18);
[Order_of_element,_Number_of_that_order]=[ [ 1, 1 ],
 [ 2, 19 ], [ 3, 2 ], [ 6, 2 ], [ 9, 6 ], [ 18, 6 ] ]
```

Poznámka. *Pro dihedrální grupy D_{2n} platí, že počet prvků řádu 2 je roven $2n+1$;*

Závěr

Cílem mé bakalářské práce bylo sepsat stručný manuál k programu GAP a demonstrovat jeho funkce na příkladech zaměřených na problematiku konečných grup.

V práci jsem se snažila zužitkovat většinu znalostí nabytých studiem nejen konečných grup, ale teorie grup celkově a v neposlední řadě základy programování. Základní znalosti z této oblasti informatiky pro mě byly při vypracovávání textu velkou výhodou.

Po dokončení práce mohu zhodnotit, že informace a znalosti, které jsem při jejím zpracovávání získala, byly a jsou pro mě velmi zajímavé a přínosné. Doufám, že zajímavá a přínosná bude pro čtenáře i má bakalářská práce.

Literatura

[1] The GAP Group: *GAP-A Tutorial*, Release 4.8.8, 2017

[2] Gallian J.: *Contemporary Abstract Algebra*, 8th Edition, University of Minnesota Duluth, BROOKS/COLE CENGAGE Learning

[3] Rainbolt J. G., Gallian J. A.: *Abstract Algebra with GAP*

[4] Rotman J.: *An introduction to the theory of groups*, Springer, 1994

[5] Hulpke A.: *Notes on Computational Group Theory*, Colorado State University, 2010

[6] Rosický J.: *ALGEBRA, Grupy a okruhy*, Brno, 2000