

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



(VRAJ) NAJŠKAREDŠIA HUDBA OD SCOTTA
RICKARDA - AKO SÚVISÍ OPAKOVANIE V HUDBE A
JEJ VNÍMANIE POSLUCHÁČMI

BAKALÁRSKA PRÁCA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**(VRAJ) NAJŠKAREDŠIA HUDBA OD SCOTTA
RICKARDA - AKO SÚVISÍ OPAKOVANIE V HUDBE A
JEJ VNÍMANIE POSLUCHÁČMI**

BAKALÁRSKA PRÁCA

Študijný program: Ekonomická a finančná matematika
Študijný odbor: 1114 Aplikovaná matematika
Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky
Vedúci práce: doc. RNDr. Beáta Stehlíková, PhD.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

PRIHLÁŠKA NA ZÁVEREČNÚ PRÁCU

Meno a priezvisko študenta: Katarína Benešová
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: (Vraj) najškaredšia hudba od Scotta Rickarda - ako súvisí opakovanie v hudbe a jej vnímanie poslucháčmi

Cieľ: Práca bude vychádzať z prednášky Scotta Rickarda na TEDxMIA. Jej štruktúra bude nasledovná: (a) Oboznámiť čitateľa s obsahom uvedenej prednášky. (b) Podrobnejšie vysvetliť používané matematické pojmy a prakticky ukázať na krátkom príklade, ako ich využil na skompovovanie "najškaredšej hudby". (c) Experimentom a jeho štatistickým vyhodnotením zistiť, či vnímanie hudby ako peknej naozaj súvisí s opakovaním tak, ako to uvádza Rickard. (d) Pokračovanie práce bude závisieť od - pozitívnej alebo negatívnej - odpovede na predchádzajúcu otázku.

Vedúci: RNDr. Beáta Stehlíková, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.

Dátum schválenia: 18.11.2015

podpis študenta

Podakovanie Moje úprimné podakovanie patrí školiteľke doc. RNDr. Beáte Stehlikovej, PhD. za jej ochotu, čas, podnetné nápady a pripomienky. Ďalej ďakujem za podporu mojej rodine a priateľom a ďakujem za ochotu všetkým, ktorí sa zapojili do experimentu.

Abstrakt

BENEŠOVÁ, Katarína: (Vraj) najškaredšia hudba od Scotta Rickarda - ako súvisí opakovanie v hudbe a jej vnímanie poslucháčmi [Bakalárska práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. RNDr. Beáta Stehlíková, PhD., Bratislava 2016, 69s.

Práca rozoberá koncept pravidelnosti v hudbe podľa prof. Scotta Rickarda a jeho pokus o vytvorenie teoreticky najškaredšej melódie. Popisuje spôsob matematickej interpretácie hudby a vytvorenie Rickardovej melódie pomocou Costasovho poľa a Golombovho pravítka. Poskytuje programy na generovanie, analyzovanie a testovanie melódií s pravidelnosťou aj bez nej. Jej cieľom je skúmať, či efekt absencie pravidelnosti v melódii súvisí s hodnotením od poslucháča.

Kľúčové slová: teória čísel, Golombovo pravítko, Costasovo pole, pravidelnosť v hudbe, melódia, Quade test, χ^2 test, štatistika

Abstract

BENEŠOVÁ, Katarína: (Allegedly) the ugliest music by Scott Rickard - how relates repetition in music with its perception by listeners [Bachelor Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: doc. RNDr. Beáta Stehlíková, PhD., Bratislava 2016, 69p.

The thesis deals with the concept of repetition in music according to prof. Scott Rickard and his aim to create ugliest piece of music. It describes the mathematical interpretation of music and creating Rickard's melody using Golomb ruler and Costas array. It contains programs for generating, analysing and testing melodies both with and without repetition. The goal of this thesis is to examine if the of absence of repetition in a melody has effect on listener's rating.

Keywords: number theory, Golomb ruler, Costas array, patterns in music, melody, Quade test, χ^2 test, statistics

Obsah

Úvod	9
1 Rýchlokurz hudobnej teórie pre matematikov	11
2 Ako vytvoriť dokonale škaredú hudbu	17
2.1 Golombovo pravítko	17
2.2 Costasovo pole	20
3 Generovanie štruktúr bez pravidelnosti	28
3.1 Generovanie Costasových polí	28
3.2 Generovanie Golombových pravítok	29
4 Je Rickardova hudba skutočne najškaredšia?	34
4.1 Tvorba melódií a experiment	35
4.2 Súvislosť s hudobným vzdelaním	45
4.3 Miera pravidelnosti v náhodnej a klasickej melódii	47
Záver	52
Literatúra	54
Príloha A	56
A.1 Zdrojové kódy v jazyku Maxima	56
A.2 Zdrojové kódy v jazyku R	56
A.3 Zdrojové kódy v jazyku MiniZinc	64
Príloha B	66
B.1 Výstupy programu hudba.r	66
Príloha C	69

C.1 Formulár k experimentu	69
--------------------------------------	----

Úvod

Keby sme sa opýtali skladateľa z 18. storočia na základný princíp skladania hudby, odpovedal by podľa autorky jednoducho: hudbu skladáme tak, aby bola príjemná ľudskému uchu. Pri skladaní hudby máme k dispozícii tóny rôznej výšky a časové škálovanie, teda rytmus. Tieto základné prvky využívame na vytvorenie hudby podľa svojich subjektívnych predstáv tak aby sa páčila poslucháčovi. Na jednej strane od hudby vyžadujeme istú pravidelnosť, ktorá prináša príjemne znejúce súzvuky a postupnosť tónov. Na druhej strane nás na hudbe oslovuje originalita. Zaujme nás neštandardná konštalácia tónov, ktorú sme ešte nepočuli. Skladateľ sa teda vo všeobecnosti pohybuje medzi dvomi protipólmi: pravidelné a známe princípy verzus neštandardná kompozícia.

Tieto dva princípy sa dajú využiť celkom jednoducho. Autorka odpozorovala, že klasické skladby majú často jeden centrálny motív, teda melódiu, ktorá sa v skladbe opakuje v rôznych obmenách. Skladateľ svoj motív väčšinou prezradí na začiatku skladby. Keď sa motív v skladbe objaví znovu, už ho poznáme a máme očakávanie ako bude melódia pokračovať. Toto očakávanie v nás skladateľ (podľa [11, s.109-110]) vyvoláva za účelom naplniť ho alebo naopak porušiť. Naplnenie očakávania v nás vyvolá príjemný pocit pravidelnosti a niečoho známeho. Porušením očakávania vznikne niečo prekvapivé, nové, čo spraví skladbu zaujímavejšou.

Aby sa očakávanie vôbec dalo vyvolať je potrebné zopakovať časť skladby. Skladateľovým kľúčovým nástrojom pri tvorbe je teda opakovanie, resp. istá pravidelnosť (spomenuté v [9]). Touto myšlienkou sa zaoberal profesor Scott Rickard z Univerzity v Dubline. Položil si otázku, ako by znela hudba, v ktorej sa nedá nájsť opakovanie žiadneho druhu. Použitím diskkrétnej štruktúry menom Costasovo pole takúto špeciálnu hudbu skutočne vytvoril a odprezentoval na konferencii TEDxMIA v septembri 2011 (video z prednášky: [9]). Rickard v [9] tvrdí: ak opakovanie a pravidelnosť implikujú

peknú hudbu, potom úplná absencia opakovaní znamená teoreticky najškarredšiu hudbu. Táto téma je zaujímavou aplikáciou diskkrétnej matematiky, ktorej sa prof. Rickard intenzívne venuje. Spolu s Konstantinosom Drakakisom vydali množstvo článkov z tejto oblasti¹. Pre túto prácu bude podstatný ich článok [2] so zameraním na Costasove polia a Golombove pravítka.

V prvej kapitole sa budeme venovať matematickej interpretácii niektorých hudobných pojmov, zavedieme spôsob prepisu melódie do vektorov výšok a dĺžok tónov. V druhej kapitole vysvetlíme dva pojmy z diskkrétnej matematiky, Golombovo pravítko a Costasovo pole a pozrieme sa na niektoré ich vlastnosti. Objasníme matematiku skrytú za skladbou prof. Rickarda a tiež spôsob, akým túto špeciálnu skladbu vytvoril. V tretej kapitole využijeme vlastnosti Costasovho poľa a Golombovho pravítka na ich generovanie. V poslednej kapitole vytvoríme melódiu bez pravidelnosti a zodpovieme otázku či je skutočne škaredšia ako náhodné melódie realizovaním praktického experimentu. Pozrieme sa, či hodnotenie melódií závisí od hudobného vzdelania poslucháča a na záver porovnáme pravidelnosť v náhodne vygenerovaných melódiách s niekoľkými klasickými skladbami.

¹Kompletný zoznam publikácií je dostupný na <http://eeme.ucd.ie/~kdrakaka/documents.htm>

Kapitola 1

Rýchlokurz hudobnej teórie pre matematikov

Našou motiváciou ku skúmaniu štruktúr v hudbe je pokus o vytvorenie teoreticky najškaredšej hudby vôbec. V roku 2011 sa touto úlohou zaoberal profesor Scott Rickard z univerzity v Dubline. Výsledky jeho práce a experimentov predstavil na konferencii TEDxMIA (pozri [9]). Rickard tam prezentoval sonátu pre klavír, ktorú vytvoril pomocou matematickej štruktúry zvanej Costasovo pole.

Rickardove úvahy sú založené na hypotéze vyslovenej v [9], ktorá stanovuje, čo sa nám na hudbe páči:

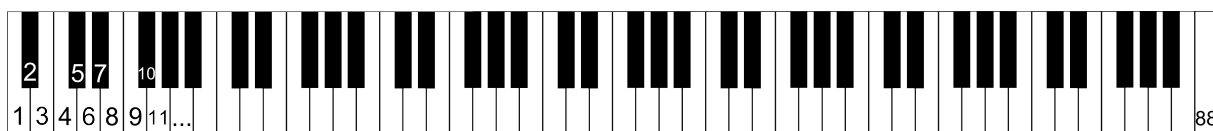
Hypotéza 1. *Na hudbe sa nám v princípe páči pravidelnosť, resp. zopakovanie štruktúry tónov. Autor týmto vyvolá očakávanie s cieľom naplniť ho, alebo naopak porušiť.*

Táto hypotéza sa intuitívne javí ako celkom rozumná. Rickarda priviedla na myšlienku, že najškaredšia hudba na svete by potom mala byť hudba, v ktorej neexistuje žiadna pravidelnosť. Ako sme už spomínali v úvode, na dosiahnutie akejkoľvek pravidelnosti je nutné zopakovať postupnosť tónov. Rickard sa rozhodol vytvoriť najškaredšiu sonátu pre klavír ako skladbu, v ktorej sa nevyskytuje žiadne opakovanie. Za účelom odstránenia pravidelnosti zabránil Rickard dvom druhom opakovania: rytmickému a výškovému.

Skôr, než sa budeme venovať Rickardovej neštandardnej kompozícii, definujme niekoľko základných pojmov spracovaných podľa [3, kap. 3, 4 a 7].

Kľúčovým objektom tejto práce, ako aj myšlienok Scotta Rickarda je melódia. Me-

lódia je postupnosť tónov, ktorá je daná poradím, výškou a dĺžkou jednotlivých tónov (pozri [7]). Ak zachováme poradie, stačí popísať výšku a dĺžku tónov. Na tento účel použijeme klasický 88-klávesový klavír, ktorého klávesy očísľujeme 1 až 88 od najhlbšej po najvyššiu (pozri Obr. 1.1). Pomlčky, teda prestávky medzi tónmi, budeme značiť nulou.



Obr. 1.1: Označenie kláves klasického klavíra

Biele klávesy a tóny, ktoré vydávajú, značíme písmenami C, D, E, F, G, A, H. Čierne klávesy zodpovedajú tónom o pol tónu vyšším alebo nižším ako biele klávesy. Medzi dvojicami tónov E-F a tiež H-C je výškový rozdiel jeden poltón, medzi ostatnými dvojicami C-D, D-E, F-G, G-A, A-H je to celý tón (dva poltóny). Na klavíri túto zákonitosť pekne vidno: medzi dvojicou tónov, ktorých rozdiel je celý tón, sa nachádza jedna klávesa (biela alebo čierna). Dve susedné klávesy majú rozdiel jeden poltón.

Zvýšenie tónu o poltón reprezentuje symbol $\sharp$ a prípona *-is*. Napríklad zvýšené C značíme $C\sharp$ a čítame cis. Zníženie tónu o poltóna dosiahneme pridaním symbolu $\flat$ a prípony *-es*. Napríklad $D\flat$ voláme des. Všimnime si, zvýšenie a zníženie o poltón môže viesť k rovnakému tónu s dvomi názvami. Napríklad pre C a D s rozdielom jeden celý tón označujú $C\sharp$ a $D\flat$ v skutočnosti ten istý tón. Podobne napr. $H\sharp=C$.

Na klavíri sa opakuje postupnosť tónov, ktorú voláme oktáva. Ide o postupnosť kláves biela, čierna, biela, čierna, biela, biela, čierna, biela, čierna, biela, čierna, biela, resp. základná postupnosť tónov C, $C\sharp$, D, $D\sharp$, E, F, $F\sharp$, G, $G\sharp$, A, $A\sharp$, H. Potom nasleduje opäť C, ktoré je o oktávu vyššie. Číslo klávesy pre daný tón vypočítame ako $3 + p + 12k$, kde p je poradové číslo tónu v základnej postupnosti a k je číslo oktávy (musíme pripočítať 3, pretože klavír začína klávesami A, $A\sharp$, H). Napr. klávesy č. $3 + 1 + 12k$, $k = 0, 2, \dots, 6$ sú tóny C v rôznych oktávach, pričom oktávu značí číslo k . Oktávy sme teda označili nasledovne: klávesy 1,2,3 sú z -1.oktávy, potom nasleduje 7 celých oktáv a posledná klávesa č. 88 je tón C zo začínajúcej 8.oktávy. Pri prepise nôt do zodpovedajúcich kláves klavíra teda stačí identifikovať výšku tónu a oktávu v ktorej sa nachádza. Tóny budeme značiť ako T_k , kde T je tón zo základnej postupnosti

a k je číslo oktávy (napr. D_4).

Na Obr. 1.2 uvádzame štandardné značenie dĺžok tónov v poradí celá, polová, štvrtová, osminová a šestnástinová nota, resp. pomlčka¹. Pri štvrtových a kratších notách a pomlčkách platí, že pridanie „zástavky“ napravo znamená skrátenie noty na polovičnú dĺžku. Viacero takýchto nôv rovnakej dĺžky môžeme spojiť čiarami, ktorých počet je rovnaký ako počet zástavok nespojených nôv (pozri spojenie šestnástinových nôv v melodii na Obr. 1.4).



Obr. 1.2: Dĺžky tónov: zľava polová, štvrtová, osminová a šestnástinová nota a pomlčka.

Základnou jednotkou merania dĺžok je štvrtová nota. Dĺžka noty, resp. pomlčky sa skracuje na polovičnú smerom zľava doprava. Polová nota trvá vždy dvojnásobok štvrtovej, celá nota dvojnásobok polovej. Osminová nota trvá naopak polovicu štvrtovej a šestnástinová polovicu osminovej. Analogicky fungujú pomlčky.

So skracovaním nôv sa dá pokračovať na dvaatricatinovú a štyriašesťdesiatinovú notu. Ďalšie dĺžky dostaneme spojením nôv rovnakej výšky do tzv. ligatúr (Obr. 1.3 v prvom takte). Dĺžka ligatúry je súčet dĺžok spojených nôv. Špeciálnym tipom ligatúry je ligatúra s notou polovičnej dĺžky. Značíme ju skrátenou bodkou za notou a takáto nota trvá 1,5-násobok svojej dĺžky (Obr. 1.3 v druhom takte).



Obr. 1.3: Príklad ligatúry, ktorá trvá $3\frac{1}{2}$ štvrtovej noty (1. takt). Osminová nota s bodkou, ktorá trvá $\frac{3}{4}$ štvrtovej noty (2. takt).

Pri skladbe býva uvedené tempo v tvare $\text{♩} = n$, napríklad $\text{♩} = 120$. Toto číslo udáva akú časť minúty sa má hrať štvrtová nota, v tomto prípade $\frac{1}{120}$ minúty, teda 0,5

¹Šestnástinová nota a pomlčka sú na obrázku dvakrát, pretože že súčet dĺžok nôv v takte musí byť taký, aký udáva predznamenanie (dvojcíslo na začiatku riadku). Číslo $\frac{a}{b}$ značí, že jeden takt trvá a nôv „dĺžky“ $\frac{1}{b}$. Napríklad predznamenanie $\frac{4}{4}$ hovorí, že jeden takt trvá štyri štvrtové noty.

sekundy. V závislosti od toho dopočítame dĺžky ostatných nôt.

Pomocou takejto interpretácie výšok a dĺžok tónov vieme notový text jednoducho prepísať na dva číselné vektory ekvivalentné pôvodnej melódii.

Definícia 1. *Melódia*

Melódiu charakterizujeme vektormi v a d , kde

- *v je vektor výšok tónov, resp. vektor celých čísel z intervalu $[0,88]$, kde čísla 1 až 88 zodpovedajú klávesám na klavíri a nula znamená pomlčku.*
- *d je vektor dĺžok tónov a pomlčiek v sekundách*

Prepis melódie do vektorov u a v si ukážeme na príklade známej melódie Pre Elišku od Ludwiga van Beethovena (pozri [1]).

Príklad 1. *Melódia prepísaná pomocou vektorov v a d*

Pre Elišku

Ludwig van Beethoven



Obr. 1.4: Úryvok zo skladby Pre Elišku

Vektor výšok v : Prvá nota E sa nachádza v 4. oktáve, $3+5+4 \cdot 12=56$, takže táto nota zodpovedá 56. klávese klavíra. Týmto spôsobom dopočítame klávesy pre ostatné noty a pomlčky označíme nulou: $v = [56, 55, 56, 55, , 56, 51, 54, 52, 49, 0, 40, 44, 49, 51, 0, 44, 48, 51, 52]$ *Vektor dĺžok d :* tempo skladby je 70 štvrtových nôt za minútu. Jedna štvrtová nota trvá $\frac{1}{70}$ minúty, teda $\frac{6}{7} \doteq 0,857$ sekundy. Dopocítame dĺžky tónov a polmčiek: osminová nota a polmčka $= \frac{1}{2}$ štvrtovej $= \frac{3}{7} \doteq 0,429$ sekundy, šestnástinová nota a pomlčka $= \frac{1}{2}$ osminovej $\doteq 0,215$ sekundy. Vektor výšok v v sekundách bude $d = [0.215, 0.215, 0.215, 0.215, 0.215, 0.215, 0.215, 0.215, 0.429, 0.215, 0.215, 0.215, 0.215, 0.429, 0.215, 0.215, 0.215, 0.429]$

V tejto práci sa budeme zaoberať skladbami, v ktorých znie naraz vždy len jeden tón, tzv. monofóniami. Skladby obsahujúce súzvky značne komplikujú interpretáciou hudby ako postupnosti. Melódiu chápeme ako monofonickú skladbu.

Výškové vzdialenosti (v hudobnej náuke *interval*) medzi tónmi sa štandardne merajú v poltónoch. K predstave opäť pomôže klavír. Vzdialenosť medzi dvoma susednými klávesami (vrátane čiernych) je jeden poltón. V definícii melódie máme výšky tónov vyjadrené číslami, ktoré zodpovedajú klávesám na klavíri, čo umožňuje popísať meranie vzdialenosti tónov nasledovným spôsobom:

Definícia 2. *Interval dvoch tónov*

Vzdialenosť, alebo interval i -teho a j -teho tónu vo vektore v je rozdiel $v_i - v_j$.

Pre $v_i > v_j$ je interval dvoch tónov vo v počet poltónov ležiacich medzi nimi. Ak $v_i < v_j$, ich interval je mínus počet poltónov. V tejto práci v súlade s Rickardovou prácou považujeme intervaly k a $-k$ za rôzne.

Rickard sa snažil vytvoriť melódiu, ktorá ľudskému uchu znie čo najhoršie. Zvolil štruktúru, v ktorej sa nedá nájsť žiadne opakovanie alebo pravidelnosť. V [9] pre ju označuje pojmom *pattern-free structure*¹. Výsledkom je melódia, v ktorej sa nedá nájsť opakovanie z hľadiska výšok ani dĺžok tónov. V prípade melódie to znamená nasledovné:

Definícia 3. *Melódia bez opakovania intervalov*

Melódiou bez opakovania intervalov nazývame melódiu s takouto vlastnosťou: Ak sa v melódii raz vyskytne dvojica tónov s intervalom k poltónov, už v nej nenájdeme inú dvojicu s intervalom k .

Všimnime si, že v definícii nehovoríme len o dvoch po sebe idúcich tónoch, ale o ľubovoľnej dvojici tónov. Tak to je aj v Rickardovej sonáte.

Melódia Pre Elišku z Príkladu 1 nie je melódiou bez opakovania intervalov. Vidno to na 16×16 matici intervalov. Z vektora výšok vynecháme pomlčky (nuly) a napočítame intervaly všetkých dvojíc tónov: 2. tón -1 . tón, 3. $-1, \dots, 17$. -1 . bude 1. stĺpec matice, 3. $-2, \dots, 17$. -2 . bude 2. stĺpec až 17. -16 . bude posledný stĺpec.

¹Z angl. pattern - vzor, schéma, pravidelnosť; pattern-free - nepravidelný, bez závislosti; structure - štruktúra, usporiadanie

Tabuľka 1.1: Matica intervalov M kde $M_{r,s} = v[r+1] - v[s]$, $r = 1, \dots, 16$, $s = 1, \dots, 16$

-1																	
0	1																
-1	0	-1															
0	1	0	1														
-5	-4	-5	-4	-5													
-2	-1	-2	-1	-2	3												
-4	-3	-4	-3	-4	1	-2											
-7	-6	-7	-6	-7	-2	-5	-3										
-16	-15	-16	-15	-16	-11	-14	-12	-9									
-12	-11	-12	-11	-12	-7	-10	-8	-5	4								
-7	-6	-7	-6	-7	-2	-5	-3	0	9	5							
-5	-4	-5	-4	-5	0	-3	-1	2	11	7	2						
-12	-11	-12	-11	-12	-7	-10	-8	-5	4	0	-5	-7					
-8	-7	-8	-7	-8	-3	-6	-4	-1	8	4	-1	-3	4				
-5	-4	-5	-4	-5	0	-3	-1	2	11	7	2	0	7	3			
-4	-3	-4	-3	-4	1	-2	0	3	12	8	3	1	8	4	1		

Rovnaké čísla v matici zodpovedajú dvojiciam tónov s rovnakým intervalom. Napríklad -1 na pozícii $[1,1]$ znamená, že druhý tón je o jeden poltón nižší ako prvý. Nuly označujú dvojice tónov s rovnakou výškou.

Okrem nepravidelnosti intervalov vyžadoval Rickard nepravidelnosť dĺžok tónov. Tóny sú reprezentované časovým okamihom kedy začínajú znieť, pričom znejú až do začiatku ďalšieho tónu, prípadne pomlčky.

Definícia 4. *Melódia bez opakovania dĺžok*

Melódiou bez opakovania dĺžok tónov nazývame melódiu s takouto vlastnosťou: Ak sa v melódii raz vyskytne dvojica tónov, ktorých rozdiel časov, kedy začínajú znieť, je t sekúnd, už v nej nenájdeme inú dvojicu s rozdielom t .

V takejto melódii musí mať každý tón inú dĺžku, pretože definícia platí aj pre dvojice po sebe idúcich tónov. Na vektore dĺžok d melódie Pre Elišku z Príkladu 1 vidíme, že rovnaká dĺžka 0,215 sekundy sa vyskytuje veľa krát, takže nejde o melódiu bez opakovania dĺžok.

Kapitola 2

Ako vytvoriť dokonale škaredú hudbu

V tejto kapitole rozoberieme matematickú stránku Rickardovej populárnej prednášky [9]. Rozoberieme pojmy Golombovo pravítko a Costasovo pole, pomocou ktorých vytvoril spomínanú sonátu. Ukáže sa, že ich použitie vedie k melódii bez opakovania výšok aj dĺžok tónov. Ďalej sa budeme venovať spôsobom generovania Costasovho poľa pomocou primitívneho koreňa modulo n . Na záver ukážeme, že špeciálnym rozložením Golombovho pravítka sa dá vytvoriť Costasovo pole.

2.1 Golombovo pravítko

Štandardné pravítko dĺžky n na sebe má $n + 1$ značiek a vzdialenosti medzi značkami sú konštantné. Pravítko sa vyznačuje jednoduchou vlastnosťou - vieme ním odmerať všetky vzdialenosti od 1 po n . Predstavme si, že by sme chceli pravítko, ktoré zachováva túto dobrú vlastnosť, ale má minimálny počet značiek.

V ďalších úvahách budeme chápať pravítko ako m -ticu čísel zoradených od najmenšieho po najväčšie, ktoré predstavujú čísla na značkách. Za dĺžku pravítka budeme považovať maximálnu vzdialenosť, ktorá sa ním dá namerať, teda rozdiel čísla na poslednej a prvej značke. Golombovo pravítko je špeciálne tým, že vzdialenosti, ktoré sa ním dajú odmerať, sú všetky rôzne. Formálne ho zdefinujeme podľa [2].

Definícia 5. *Golombovo pravítko*

Nech $m, n \in \mathbb{N}$ a $m \leq n + 1$. Golombovým pravítkom dĺžky n s m značkami nazývame vektor prirodzených čísel g s nasledvnou vlastnosťou:

$$\forall i, j, k, l \in [1, \dots, m] : g[i] - g[j] = g[k] - g[l] \iff i = k \wedge j = l$$

Poznámka 1. V literatúre sa vyskytujú aj Golombove pravítka rozsahu $[0, n]$. V tejto práci budeme konzistentní s [2], teda pravítka dĺžky n začínajú značkou 1 a končia značkou $n + 1$.

Ideálne by bolo, aby Golombovo pravítko dlhé n vedelo namerať namerať všetky dĺžky od 1 po n . Takéto pravítko možno skonštruovať napríklad pre $n = 6$:

Príklad 2. Golombovo pravítko dĺžky 6 so štyrmi značkami vygenerované programom *golomb.mnz* (pozri prílohu A): $[1, 2, 5, 7]$. Dajú sa ním namerať vzdialenosti 1, 4, 6, 3, 5 a 2. Sú všetky rôzne, takže ide o Golombovo pravítko. Navyše dĺžka pravítka je $7-1=6$ a nameriame ním všetky vzdialenosti od 1 po 6.

Pre dĺžku $n = 6$ sme našli pravítko s $m = 4$ značkami, ktorým sa dajú namerať všetky dĺžky od 1 po n . Takéto pravítka nazývame dokonalé Golombove pravítka. Nanešťastie, najväčší počet značiek, pre ktoré vieme skonštruovať dokonalé Golombovo pravítko je $m = 4$. Pre väčšie m budeme musieť upustiť od snahy odmerať pravítkom všetky vzdialenosti od 1 po n . Napríklad pre $m = 8$ potrebujeme dĺžku aspoň $n = 35$ a niektoré vzdialenosti od 1 po 35 na pravítku chýbajú (napr. vzdialenosť 16):

Príklad 3. Golombovo pravítko dĺžky 34 s ôsmimi značkami vygenerované programom *golomb.mnz* (pozri prílohu A): $[1, 2, 5, 10, 16, 23, 33, 35]$

Skutočnosť, že pre $m > 4$ musia niektoré vzdialenosti chýbať zdôvodníme v nasledujúcej vete inšpirovanej [13].

Veta 1. Pre $m > 4$ neexistuje dokonalé Golombovo pravítko.

Dôkaz: Ukážeme sporom. Pokúsme sa také pravítko skonštruovať.

Na Golombovom pravítku s m značkami nameriame $\binom{m}{2}$ rôznych vzdialeností. Predpokladáme, že sú to práve vzdialenosti od 1 po n pre nejaké $n \geq m$. Potom nutne ide o pravítko dĺžky $n = \binom{m}{2}$. Ak by bolo kratšie, mali by sme menej než $\binom{m}{2}$ rôznych vzdialeností, niektorá by sa teda zopakovala. Ak by bolo dlhšie, existovala by vzdialenosť medzi 1 a n , ktorú na pravítku nenájde.

Na pravítku preto určite budú značky 1 a $\binom{m}{2} + 1 = n + 1$ predstavujúce začiatok a koniec pravítka. Ďalej sa na pravítku musí byť dvojica značiek s vzdialenosťou $n - 1$. Máme dve možnosti: $[1, n, n + 1]$ alebo $[1, 2, n + 1]$. Vďaka ich symetrii si bez ujmy na všeobecnosti môžeme vybrať $[1, 2, n + 1]$.

Pre vzdialenosť $n - 2$ máme tri možnosti: $[1, 2, n - 1, n + 1]$, $[1, 2, n, n + 1]$ a $[1, 2, 3, n + 1]$. Vyhovuje jedine prvá, keďže zvyšné dve duplikujú vzdialenosť 1 .

Vzdialenosť $n - 3$ už na pravítku máme medzi 2 a $n - 1$.

Naše pravítko aktuálne vyzerá $[1, 2, n - 1, n + 1]$. Vzdialenosť $n - 4$ môžeme získať pridaním značky na $n - 3$, $n - 2$, 3 alebo 5 . Duplikáty nevytvára jedine $[1, 2, 5, n - 1, n + 1]$. Pre $m = 4$, resp. $n = 6$ dostávame pravítko z Príkladu 1.1. Pre $m = 5$, resp. $n = 10$ dostávame pravítko $[1, 2, 5, 9, 11]$, v ktorom je zduplikovaná vzdialenosť 4 .

Pre $m > 5$ pokračujeme s pravítkom $[1, 2, 5, n - 1, n + 1]$. Vzdialenosť $n - 5$ dosiahneme pridaním značky na $n - 4$, $n - 3$, n , 4 alebo 6 , avšak všetky vedú k zduplikovaniu vzdialeností $1, 2$, alebo 3 . Preto dokonalé Golombovo pravítko s viac ako štyrmi značkami neexistuje. ■

Golombovo pravítko ako také sa dá skonštruovať veľmi jednoducho, napríklad $[2^0, 2^1, \dots, 2^m]$ bude určite Golombovo pravítko. Vzdialenosti medzi po susednými značkami sú $1, 2^1, 2^2, \dots, 2^{m-1}$, takže vzdialenosť od k -tej po značku číslo $k+1$ je 2^k , čo je viac ako najdlhšia možná vzdialenosť po k -tu značku $1 + 2^1 + 2^2 + \dots, 2^{k-1} = 2^k - 1$. Keďže medzi dve susedné značkami sa zmestí celá vzdialenosť od začiatku pravítka po menšiu značku, na pravítku nemôžeme nájsť dve dvojice rovnako vzdialených značiek. Formálne, majme 4 značky $2^a, 2^b, 2^c, 2^d$, $1 \leq a < b < c < d \leq m$. Ak $2^c - 2^a = 2^d - 2^b$, resp. $2^a(2^{c-a} - 1) = 2^b(2^{d-b} - 1)$, tak nutne $2^a = 2^b$, pretože a a b nie sú nič iné ako počty dvojiek v prvočíselnom rozklade čísla $2^a(2^{c-a} - 1) = 2^b(2^{d-b} - 1)$. Prvé dve značky by museli byť totožné, čo je spor. Pre väčšie m by ale takéto pravítka museli byť veľmi dlhé, preto ich chcele konštruovať efektívnejšie. Konštrukcia Golombovho pravítka je v princípe optimalizačná úloha:

- pre danú dĺžku n chceme minimálny počet značiek, alebo
- daný počet značiek m chceme čo najmenšiu dĺžku n

Profesor Rickard použil Golombovo pravítko ako vektor dĺžok d , čím zaručil, že jeho sonáta bude melódiou bez opakovania dĺžok V jeho sonáte sa nenachádzajú pomlčky,

preto i -ty tón začína v čase $d[i]$, znie $d[i + 1] - d[i]$ časových jednotiek (napr. sekúnd) a v čase $d[i + 1]$ začína znieť ďalší. Pravítkom s m značkami teda môžeme určiť dĺžky pre melódiu s $m - 1$ tónmi.

2.2 Costasovo pole

Po vyriešení otázky rytmiky sa pozrime na problematiku získania vektora výšok tónov v . Rickard chcel využiť plný rozsah klavíra, chcel teda melódiu dlhú 88 tónov a chcel aby sa v nej neopakovali výšky tónov. Prirodzeným riešením by bolo aj pre vektor v použiť Golombovo pravítko. Prakticky to je ale nemožné, pretože pre $m = 88$ by sme podľa [6] potrebovali pravítko dlhé aspoň $n = 6745$, čo značí 6745 kláves. Na klavír s 88 klávesami, resp. pravítko s dĺžkou najviac 88 by sa podľa [6] zmestilo najviac 12 tónov takejto melódie.

Riešením k zachovaniu 88-tónovej sonáty je dvojrozmerná analógia Golombovho pravítka, Costasovo pole. Ako hovorí Rickard vo videu, Costasovo pole si môžeme predstaviť ako štvorcovú mriežku vyplnenú bodkami tak aby platilo:

- v každom riadku aj slpci je práve jedna bodka
- všetky vektory tvorené dvojicami bodiek sú navzájom rôzne

Profesor Rickard a Dr. Drakakis vo svojej práci [2] definujú Costasovo pole takto:

Definícia 6. *Costasovo pole*

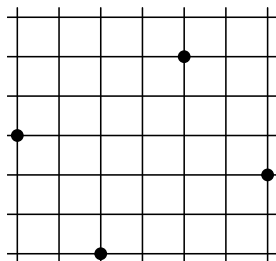
Costasovo pole veľkosti s je $s \times s$ mriežka s bodmi na mrežových bodoch umiestnenými tak, že platí:

- *na každej vodorovnej, resp. zvislej čiare leží práve jeden bod*
- *žiadne štyri body netvoria rovnobežník*
- *ak tri body označené zľava A, B, C ležia na jednej priamke, dĺžky úsečiek AB a BC sú rôzne*

Poznámka 2. Pod súradnicami bodu myslíme dvojicu $[x, y]$, kde x je x -tá vylá čiara y je y -tá vodorovná čiara, pričom začíname od ľavého dolného rohu mriežky (bod $[1, 1]$).

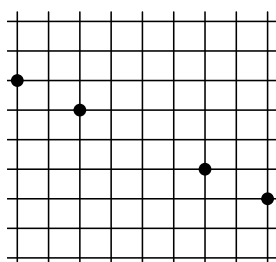
Costasovo pole môžeme tiež chápať ako kladný kvadrant súradnicovej sústavy s počiatkom v $[1, 1]$, kde čiary mriežky sú priamky kolmé na os x , resp. y prechádzajúce bodmi $[1, 1], [2, 1], \dots, [s, 1]$ resp. $[1, 1], [1, 2], \dots, [1, s]$.

Za rovnobežník budeme považovať také dve dvojice bodov, ktoré majú zhodné príslušné rozdiely x -ových a y -ových súradníc. Nech $[x_1, y_1], [x_2, y_2], [x_3, y_3], [x_4, y_4]$ sú štyri body ležiace na mrežových bodoch mriežky a nech sú zoradené zľava doprava, teda $x_1 < x_2 < x_3 < x_4$. Budeme ich považovať za rovnobežník, ak $y_2 - y_1 = y_4 - y_3$ a zároveň $x_3 - x_1 = x_4 - x_2$ (pozri Obr. 2.1). Zmysel takejto definície sa ukáže, keď si predstavíme štyri body z Obr. 2.1 ako tóny, kde y -ové súradnice sú ich výšky. Potom takýto rovnobežník vyjadruje istú pravidelnosť, resp. opakovanie. Výškový rozdiel $y_2 - y_1 = y_4 - y_3 = -3$ poltóny sa opakuje medzi dvojicou prvý-tretí a piaty-siedmy tón, ktoré sú rovnako vzdialené z hľadiska poradia: $x_3 - x_1 = x_4 - x_2 = 2$. Takáto definícia navyše sa ukáže ako praktická pri neskoršom vyhľadávaní rovnobežníkov v mriežkach.

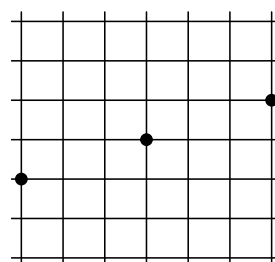


Obr. 2.1: Príklad rovnobežníka vykresleného v R programom *mriezky.r*

Všimnime si, že na základe takejto definície považujeme za rovnobežník aj takú štvoricu bodov, ktoré ležia na priamke (Obr. 2.2a). Takýto degenerovaný rovnobežník nie je rovnobežníkom v klasickom ponímaní, avšak vyjadruje pravidelnosť, preto ho budeme rovnobežník považovať. Na Obr. 2.2b je trojica bodov, ktorá nevyhovuje tretiemu bodu definície.



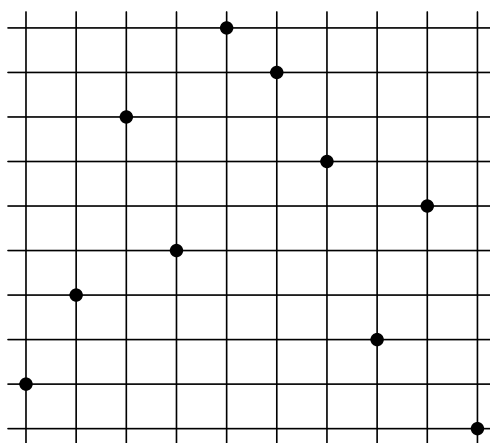
(a)



(b)

Obr. 2.2: Príklady degenerovaných rovnobežníkov vykreslených v R programom *mriezky.r*

Costasovo pole pre $s = 10$ ilustrujeme na Obr.2.3.

Obr. 2.3: Costasovo pole 10×10 vykreslené v R programom *mriezky.r*

Rickardova myšlienka za vytvorením melódie, v ktorej sa nedá nájsť opakovanie z hľadiska výšových rozdielov, je jednoduchá: ako vektor výšok tónov použijeme y -ové súradnice bodov Costasovho poľa. Otázkou je, ako takéto pole vytvoriť. Rickard

hovorí o spôsobe, akým vygeneroval Costasovo pole 88×88 v prednáške [9]: "Pole je vygenerované mocninami čísla 3: 3,9,27,81,243,... . Keď sa dostaneme nad číslo 89, čo je prvočíslo, odčítavame 89 až kým sa nedostaneme naspäť. Týmto nakoniec vyplníme celú mriežku 88×88 ." Rickard teda hovorí, že i -ty bod (bod na i -tej zvislej čiare) má y -novú súradnicu veľkosti $3^i \bmod 89$.

1. stĺpec:	$3^1 = 3$	$3 \equiv 3 \bmod 89$	prvý tón má výšku 3
2. stĺpec:	$3^2 = 9$	$9 \equiv 9 \bmod 89$	druhý tón má výšku 9
3. stĺpec:	$3^3 = 27$	$27 \equiv 27 \bmod 89$	druhý tón má výšku 27
4. stĺpec:	$3^4 = 81$	$81 \equiv 81 \bmod 89$	druhý tón má výšku 81
5. stĺpec:	$3^5 = 243$	$243 \equiv 65 \bmod 89$	druhý tón má výšku 65
6. stĺpec:	$3^6 = 729$	$729 \equiv 17 \bmod 89$	druhý tón má výšku 17
...			
88. stĺpec:	$3^{88} = 9,697 \times 10^{41}$	$3^{88} \equiv 1 \bmod 89$	posledný tón má výšku 1

Nie každá mriežka vygenerovaná spôsobom umocňovanie-modulovanie bude Costasovo pole. Výsledné pole závisí od voľby čísla, ktoré umocňujeme a prvočísla, ktorým modulujeme. Rickard vytvoril Costasovo pole, pretože číslo 3 je tzv. primitívny koreň modulo 89 a 89 je prvočíslo. Primitívny koreň k číslu n , alebo tiež primitívny koreň modulo n , je také špeciálne číslo, ktorého mocniny majú určité vlastnosti v súvislosti s n (podľa [10, s. 16]).

Definícia 7. *Primitívny koreň modulo n*

Nech $\lambda, n, c \in \mathbb{N}$. Číslo λ je primitívnym koreňom modulo n ak pre každé číslo c nesúdeliteľné s n platí:

$$c \equiv \lambda^k \bmod n$$

pre nejaké $k \in \mathbb{N}$.

Poznámka 3. Ak za n vezmeme prvočíslo p , definícia platí pre všetky $c \in \mathbb{N}$ okrem p a 1. Potom sa pre každé $c \in \mathbb{N}$ okrem p a 1 musí nájsť nejaká mocnina čísla λ taká, že λ^k a c dávajú po delení p rovnaký zvyšok. Teda platí, že ak je λ primitívny koreň modulo prvočíslo, mocniny λ musia dávať po delení p všetky zvyšky $1, 2, \dots, p-1$.

Ukážme si príklad prvočísla a jeho primitívneho koreňa:

Príklad 4. *Príklad: $\lambda = 2$ je primitívnym koreňom modulo $n = 11$*

$2^1 = 2$	$2 \equiv 2 \pmod{11}$
$2^2 = 4$	$4 \equiv 4 \pmod{11}$
$2^3 = 8$	$8 \equiv 8 \pmod{11}$
$2^4 = 16$	$16 \equiv 5 \pmod{11}$
$2^5 = 32$	$32 \equiv 10 \pmod{11}$
$2^6 = 64$	$64 \equiv 9 \pmod{11}$
$2^7 = 128$	$128 \equiv 7 \pmod{11}$
$2^8 = 256$	$256 \equiv 3 \pmod{11}$
$2^9 = 512$	$512 \equiv 6 \pmod{11}$
$2^{10} = 1024$	$1024 \equiv 1 \pmod{11}$

Zvyšok 1 sa objavil pri dvojici (89,3) aj (11,2) ako posledný. Nie je náhoda, o tejto vlastnosti prvočísel hovorí známa Malá Fermatova veta (definícia podľa [10], s. 16]):

Veta 2. Malá Fermatova veta

Pre ľubovoľné prvočíslo p a ľubovoľné $a \in \mathbb{N}$ také, že $\text{nedelí } a$ platí:

$$a^{p-1} \equiv 1 \pmod{p}$$

Budeme voliť $a = \lambda$ menšie ako p , preto zjavne $p \nmid a$. Naším najbližším cieľom je ukázať že mriežka vytvorená pomocou primitívneho koreňa je Costastovým poľom¹.

K dôkazu sú kľúčové vlastnosti mocnín primitívneho koreňa λ . V snahe zabrániť opakovaniám by sme chceli, aby bol v každom riadku Costasovho poľa najviac jeden bod. Preto je prirodzené pozrieť sa, či sa medzi príslušnými číslami $\lambda = \{\lambda^1 \pmod{p}, \lambda^2 \pmod{p}, \dots, \lambda^{p-1} \pmod{p}\}$ môžu nájsť dve rovnaké čísla (vlastnosť I.). Ďalej budeme potrebovať rozoznať koniec periódy zvyškov mocnín λ po delení p (vlastnosť II.).

Lema 1. Vlastnosti primitívneho koreňa

Nech $P_\lambda = \{\lambda^1 \pmod{p}, \lambda^2 \pmod{p}, \dots, \lambda^{p-1} \pmod{p}\}$. Preto platí:

$$I. \quad \lambda^k \not\equiv \lambda^l \pmod{p} \quad \forall k = 1, 2, \dots, p-1, l = 1, 2, \dots, p-1, k \neq l$$

¹V literatúre sa vyskytujú dôkazy o generovaní Costasových polí týmto spôsobom, avšak nie v zmysle rovnobežníkovej definície. Preto uvádzame vlastný dôkaz.

II. jediný prvok z P_λ dávajúci zvyšok 1 mod p je λ^{p-1}

Dôkaz: Ukážeme, že P_λ obsahuje všetky zvyšky po delení p okrem 0, t.j. čísla $1, 2, \dots, p-1$, každé práve raz.

Podľa Malej Fermatovej vety platí $\lambda^{p-1} \equiv 1 \pmod{p}$, zvyšok 1 dáva tiež λ^0 . Z toho vyplýva:

- (a) pre $k = p - 1$ začína nová perióda zvyškov, pre väčšie k sa budú zvyšky už len opakovať
- (b) P_λ je jednou alebo viacerými periódami zvyškov $\lambda^k \pmod{p}$

Keďže λ je primitívny koreň mod p , jedna zvyšková perióda mocnín λ po delení p musí byť tvorená všetkými číslami $1, 2, \dots, p-1$. Preto musí byť $(p-1)$ -prvková množina P_λ jednou celou periódou a teda $P_\lambda = \{1, 2, \dots, p-1\}$. Preto platí:

$$I. \lambda^k \not\equiv \lambda^l \pmod{p} \quad \forall k = 1, 2, \dots, p-1, l = 1, 2, \dots, p-1, k \neq l.$$

II. jediný prvok z P_λ dávajúci zvyšok 1 mod p je λ^{p-1} ■

Veta 3. Veta o generovaní

Nech $p \in \mathbb{N}$ je prvočíslo a $\lambda \in \mathbb{N}$ je jeho primitívnym koreňom. Potom $(p-1) \times (p-1)$ mriežka s mrežovými bodmi na súradniciach $[k, \lambda^k \pmod{p}]$, kde $k = 1, 2, \dots, p-1$ je Costasovo pole v zmysle Def. 7.

Dôkaz: Súradnice bodov takejto mriežky tvoria množinu P_λ z Lemy 1. Chceme ukázať, že takto vygenerovaná mriežka spĺňa:

1. na každej vodorovnej, resp. zvislej čiare leží práve jeden bod
 2. žiadne štyri body netvoria rovnobežník
 3. ak tri body označené zľava A,B,C ležia na jednej priamke, dĺžky úsečiek AB a BC sú rôzne
1. Vezmime si dva rôzne body z P_λ : $[k, \lambda^k \pmod{p}]$ a $[l, \lambda^l \pmod{p}]$, $k \neq l$. Zjavne neležia v rovnakom riadku. Ak by ležali v rovnakom stĺpci, potom $\lambda^k \equiv \lambda^l \pmod{p}$, čo je spor.

2. Ukážeme sporom: vytvoríme si kandidátov na rovnobežník. Nech pre nejakú štvoricu bodov $[a, \lambda^a \bmod p], [b, \lambda^b \bmod p], [c, \lambda^c \bmod p], [d, \lambda^d \bmod p], a < b < c < d < p$ platí

$$c - a = d - b$$

Ak tvoria rovnobežník, zároveň kvôli y-ovým súradniciam platí:

$$(\lambda^d \bmod p) - (\lambda^c \bmod p) \equiv (\lambda^b \bmod p) - (\lambda^a \bmod p) \quad (2.1)$$

Použitím elementárnych vlastností kongruencií upravíme rovnicu na

$$\lambda^d - \lambda^c \equiv \lambda^b - \lambda^a \bmod p \quad (2.2)$$

Pozn. situáciu pre $a = 1, b = 2, c = 8, d = 9$ ilustrujeme na Obr.2.4.

Ďalej v (2.2) vyjmeme menšie z čísel na oboch stranách:

$$\lambda^c(\lambda^{d-c} - 1) \equiv \lambda^a(\lambda^{b-a} - 1) \bmod p$$

Keďže $c - a = d - b$, dostávame

$$(\lambda^{d-c} - 1)(\lambda^c - \lambda^a) \equiv 0 \bmod p \quad (2.3)$$

Jedna zo zátvoriek je deliteľná p . Rozoberieme oba prípady:

- (a) Platí $\lambda^{d-c} - 1 \equiv 0 \bmod p$. Potom $\lambda^{d-c} \equiv 1 \bmod p$, zvyšok 1 sa teda dosahuje pre $d - c < p - 1$, čo je spor s II. vlastnosťou množiny P_λ
- (b) Platí $\lambda^c - \lambda^a \equiv 0 \bmod p$. Potom $\lambda^c \equiv \lambda^a \bmod p$, čo je spor s I. vlastnosťou množiny P_λ .

3. Opäť sporom. Nech tri body z mriežky, $[a, \lambda^a \bmod p], [b, \lambda^b \bmod p], [c, \lambda^c \bmod p], a < b < c < p$, ležia na priamke. Potom platí

$$c - b = b - a$$

Zaujíma nás platnosť výrazu

$$\lambda^b - \lambda^a \equiv \lambda^c - \lambda^b \bmod p \quad (2.4)$$

Znovu vyjmeme a dostávame

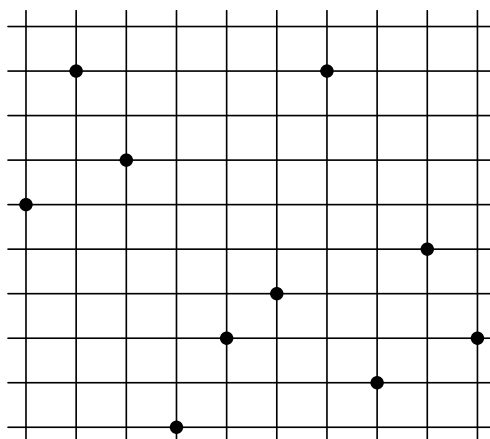
$$\lambda^a(\lambda^{b-a} - 1) \equiv \lambda^b(\lambda^{c-b} - 1) \pmod{p}$$

kde $\lambda^{b-a} - 1 = \lambda^{c-b} - 1$. Po úprave:

$$(\lambda^{b-a} - 1)(\lambda^a - \lambda^b) \equiv 0 \pmod{p}$$

Dostali sme prípad z 2., stačí len preznačiť: v prípade (a) $d' = b, c' = a$ a v prípade (b) $c' = a, a' = b$. Našli sme spor s (2.4) a tým je 3. bod vety dokázaný.

■



Obr. 2.4: Mriežka obsahujúca 5 rovnobežníkov, napr. medzi bodmi č. 1,2,8,9

V ďalšej kapitole sa pozrieme na praktickú stránku generovania Costasových polí a Golombových pravítok.

Kapitola 3

Generovanie štruktúr bez pravidelnosti

3.1 Generovanie Costasových polí

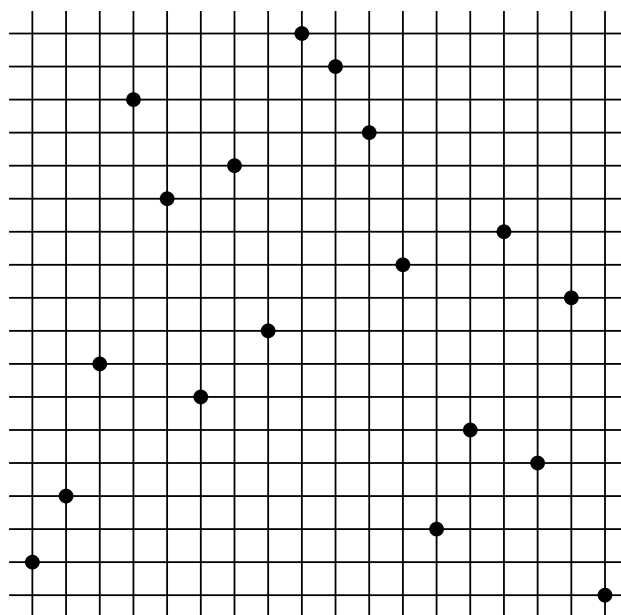
Veta o generovaní poskytuje jednoduchý spôsob ako vygenerovať Costasovo pole $(p - 1) \times (p - 1)$. Ak máme prvočíslo p a jeho primitívny koreň λ , stačí prvých $p - 1$ mocnín λ (začínajúc od λ^1) modulovať číslom p . Získame tým priamo y -ové súradnice bodov Costasovho poľa.

Prvým krokom je nájdenie primitívneho koreňa pre dané prvočíslo p , čo robíme programom *primkoren.wmx* (pozri prílohu A). Maxima sa hodí na túto úlohu, pretože nemá numerické problémy s umocňovaním a modulovaním veľkých čísel.

Program *primkoren.wmx* postupne prechádza cez čísla $2, 3, 4, \dots, \min\{p, 20\}$ a testuje, či prvých $p - 1$ mocnín aktuálneho čísla dáva po delení p všetky zvyšky $1, 2, \dots, p - 1$. Takýto rozsah sa ukázal ako postačujúci pre každé prvočíslo, ktorému sme potrebovali nájsť primitívny koreň. V prípade potreby sa dá jednoducho rozšíriť v programe *primkoren.wmx*, avšak ak chceme pridať nejaké číslo a väčšie ako p , musí platiť $p \nmid a$, kvôli Malej Fermatovej vete. Program skončí pri prvom vyhovujúcom čísle a vypíše ho. Zvyšky $\{\lambda^1 \bmod p, \lambda^2 \bmod p, \dots, \lambda^{p-1} \bmod p\}$ ukladá do 'zvysky.txt'. Napríklad pre $p = 19$ program nájde najmenší primitívny koreň $\lambda = 2$ a zvyšky $[2, 4, 8, 16, 13, 7, 14, 9, 18, 17, 15, 11, 3, 6, 12, 5, 10, 1]$.

So získanými zvyškami pokračujeme v jazyku R programom *costas.r* (pozri prí-

lohu A). Zadáme použité prvočíslo, ktorému sme našli primitívny koreň pomocou *primkoren.wmx*, napr. $p = 19$, a načítame 'zvysky.txt', ktoré ďalej považujeme za výšky. Ďalej program nastaví rozmer mriežky na $ohr = p - 1 = 18$ a vytvorí vektor poradia $por = [1, \dots, p - 1]$. Poradie a výšky spojíme do $(ohr \times 2)$ matice 'hudba' a môžeme vykresliť Costasovo pole.



Obr. 3.1: Costasovo pole pre $p = 19$ a $\lambda = 2$

3.2 Generovanie Golombových pravítok

Konštrukcia Golombovho pravítka sa oproti Costasovým poliam ukazuje byť ťažším problémom. Prvým prístupom je priamo riešiť optimalizačnú úlohu čo najmenšej dĺžky pri danom počte značiek za podmienky rôznych vzdialeností medzi značkami. Na toto je ideálny jazyk MiniZinc, ktorý funguje na princípe optimalizovania premennej pri zadávaných podmienkach. Programu *golomb.mnz* doc. Stehlíkovej (pozri prílohu A) zadáme počet značiek m a ohraničenie $nmax$ pre posledné číslo na pravítku. Ďalej zadáme

podmienky pre výstup: posledné a najväčšie číslo na pravítku je n , $n \leq nmax$, pravítko začína číslom 1 a všetky vzdialenosti medzi značkami majú byť navzájom rôzne. MiniZinc najprv nájde minimálne n spĺňajúce podmienky a vypíše príslušné čísla na pravítku (v prípade neprípustnosti zväčšíme $nmax$). S optimálnym n potom môžeme spustiť pokračovanie kódu, čoho výsledkom budú všetky rôzne pravítka pre dané m a n . Napríklad pre $m = 5$, $nmax = 50$ nájde *golomb.mnz* minimálne $n = 12$ a jediné pravítko $[1, 2, 5, 10, 12]$.

Problém s týmto prístupom sa ukazuje, keď sa pokúšame generovať Golombove pravítka väčších rozmerov. Napr. pre $m = 20$ hľadal MiniZinc pravítko 6 hodín a bezvýsledne. Pre ďalšie účely práce potrebujeme pravítka rozmerov okolo 20, preto bolo nutné zvoliť iný prístup.

Využijeme bezproblémové generovanie Costasovho poľa a získame pravítka z neho. Všimnime si, že ľubovoľná diagonála (t.j. množina bodov z mriežky na priamke prechádzajúcej cez mrežové body) je Golombovo pravítka. Ak by nebola, je to spor s tým, že Costasovo pole neobsahuje degenerované rovnobežníky. Ak chceme pravítka s aspoň 20 značkami, podľa [6] potrebujeme dĺžku aspoň $n = 283$ a teda Costasovo pole aspoň 283×283 . Najbližšie väčšie prvočíslo je 293, pomocou ktorého vygenerujeme pole 292×292 . Ukazuje sa však, že aj diagonály veľkých Costasových polí obsahujú len málo bodov. V tabuľke uvádzame počet bodov na hlavných diagonálach niektorých Costasových polí.

p	1. diagonála	2. diagonála
293	0	2
997	2	2
1459	0	2
2003	0	4
2027	0	2
2503	0	3
3037	0	4
3119	0	0
4177	4	3

Mohli by sme prehľadávať aj iné ako hlavné diagonály, avšak profesor Rickard a Dr. Drakakis ponúkajú v [2] lepší spôsob. Golombovo pravítka možno dostať špeciálnym

"zjednodušením" Costasovho popísaným. Ku polu prilepíme zdola dostatočne veľa prázdnych mrežových bodov, pod každý stĺpec nejaký násobok danej konštanty k . Potom postupne berieme stĺpce, transponujeme ich a ukladáme vedľa seba do jedného pásu. Ostáva dourčiť konštantu k , ktorej existenciu zaručuje nasledovná veta podľa [2].

Veta 4. Pásiková veta

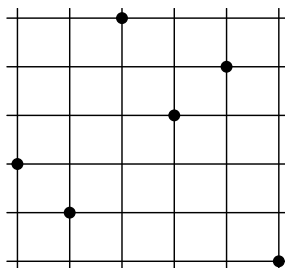
Nech $f \in \mathbb{R}^n$ je vektor y -nových súradníc bodov v Costasovom poli. Nech g je vektor daný hodnotami f a konštantou $k \in \mathbb{N}$ ako

$$g[i] = (i - 1)k + f[i] \quad \forall i = 1, \dots, n$$

Ak $k \geq 2n - 2$, potom je g Golombovo pravítko.

Najprv si transformáciu ukážme na príklade:

Príklad 5. Príklad transformácie Costasovho pola 6×6 na Golombovo pravítko so 6 značkami.



Obr. 3.2: Costasovo pole 6×6

Vektor y -ových súradníc bodov v poli je $f = [3, 2, 6, 4, 5, 1]$. Podľa Pásikovej vety $k = 2n - 2 = 10$. Pod prvý stĺpec pri dáme k prázdnych mrežových bodov, pod druhý $2k$, atď. Dopočítajme čísla na značkách pravítka:

1. značka $g[1] = 0k + f[1] = 3$
2. značka $g[2] = 1k + f[2] = 12$
3. značka $g[3] = 2k + f[3] = 26$
4. značka $g[4] = 3k + f[4] = 34$
5. značka $g[5] = 4k + f[5] = 45$
6. značka $g[6] = 5k + f[6] = 51$

Pre overenie, či sme našli Golombove pravítka, spočítajme všetkých $\frac{n(n-1)}{2} = 15$ vzdialeností medzi značkami: $[9, 23, 31, 42, 48, 14, 22, 33, 39, 8, 19, 25, 11, 17, 6]$. Sú všetky rôzne, takže máme Golombovo pravítka.

Dôkaz: Chceme ukázať, že všetky vzdialenosti na vytvorenom pravítke sú rôzne, teda podľa definície Golombovho pravítka

$$\forall i, j, v, w \in [1, \dots, m] : g[i] - g[j] = g[v] - g[w] \iff i = v \wedge j = w$$

V reči vektora f vyzerá ľavá strana ekvivalencie

$$(i - 1)k + f[i] - (j - 1)k - f[j] = (v - 1)k + f[v] - (w - 1)k - f[w]$$

Po úprave

$$f[i] - f[j] - (f[v] - f[w]) = k(j - i + v - w)$$

Chceme, aby táto rovnica mala riešenie jedine v špeciálnom prípade $i = v \wedge j = w$. V takom prípade je zátvorka $(j - i + v - w)$ nulová a bez ohľadu na k je rovnosť splnená.

Ak je zátvorka $(j - i + v - w)$ nenulová, chceme znemožniť platnosť rovnice. Docielime to tým, že nastavíme konštantu k tak aby bola pravá strana nad maximálnou alebo pod minimálnou hodnotou ľavej.

Predpokladajme $j - i + v - w \neq 0$ a pozrime sa na rozdiel $f[i] - f[j] - (f[v] - f[w])$. Ide o y-ové súradnice $n \times n$ Costasovho poľa, takže rozdiely $f[i] - f[j]$ a $f[v] - f[w]$ sú z množiny $\{-(n-1), -(n-2), \dots, -1, 1, 2, \dots, n-1\}$. Prípady $f[v] - f[w] = n-1$ a zároveň $f[i] - f[j] = n-1$ nemôže nastať, pretože by to znamenalo dva body v prvom aj n -tom riadku. Pravá strana teda nadobúda hodnoty ohraničené zdola $-(2n-3)$ a zhora $2n-3$. Ak zvolíme $k = 2n-2$, rovnica nemôže mať riešenie, pretože $|j - i + v - w| \geq 1$ a teda $|k(j - i + v - w)| > 2n-3$. ■

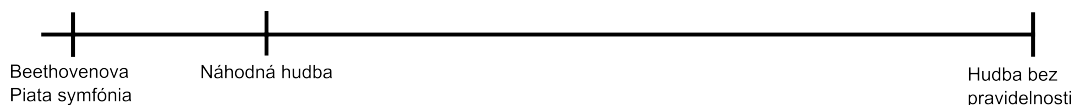
Hodnota $k = 2n - 2$ zaručuje Golombovo pravítko, nemusí však byť najmenšia možná a pravítko bude zbytočne dlhé. Pri praktickom prepise Costasovho pola na Golombovo pravítko budeme hľadať najmenšie k pre ktoré sú všetky vzdialenosti na pravítku rôzne. Robí to funkcia $COSTAS_GOLOMB(n, vysky)$, ktorú volá program *hudba.r* (pozri prílohu A), do ktorého treba zadať prvočíslo p a načítať výšky vypočítané v Maxime (*primkoren.wmx*). Program rozloží Costasovo pole na pravítko podľa Pásikovej vety a nájde minimálne k .

V ďalšej kapitole na základe týchto poznatkov vygenerujeme melódie bez opakovania výšok a dĺžok tónov.

Kapitola 4

Je Rickardova hudba skutočne najškaredšia?

Profesor Rickard v [9] tvrdí, že melódia bez opakovania výšok a dĺžok tónov je ďaleko škaredšia ako napríklad náhodne vygenerovaná melódie. Presnejšie, hovorí: „*So if we think about random music as being just random notes here, and over here is somehow Beethoven's 5th in some kind of pattern, if we wrote completely pattern-free music, it would be way out on the tail. In fact, the end of the tail of music would be these pattern-free structures. ... This is supposed to be the world's ugliest piece of music*“. Pri týchto slovách znázorňuje Rickard rukami pozíciu druhov hudby na pomyslenej úsečke približne takto:



V tejto kapitole urobíme praktický experiment pre overenie alebo vyvrátenie tohto tvrdenia a pozrieme sa či výber najškaredšej melódie závisí od hudobného vzdelania. Na záver porovnáme mieru pravidelnosti v melódiách použitých v experimente s úryvkom z klasickej skladby.

4.1 Tvorba melódií a experiment

Vytvoríme melódiu bez opakovania výšok a dĺžok tónov. Vektory v a d vygenerujeme a prepíšeme do formy pre zápis do nôt pomocou programu *hudba.r*, pričom zvolíme $p = 23$. Jeho primitívny koreň zistíme programom *primkoren.wmx*, je to $\lambda = 5$, a príslušný vektor v načítame do *hudba.r*. Zodpovedajúce Costasovo pole je dané súradnicami bodov: x-ové súradnice: $[1, 2, \dots, 22]$, y-ové súradnice (vektor v): $[5, 2, 10, 4, 20, 8, 17, 16, 11, 9, 22, 18, 21, 13, 19, 3, 15, 6, 7, 12, 14, 1]$.

Pole ďalej rozložíme podľa Pásikovej vety funkciou *COSTAS_GOLOMB(22, v).r* a dostávame Golombovo pravítko dĺžky 22: $[5, 37, 80, 109, 160, 183, 227, 261, 291, 324, 372, 403, 441, 468, 509, 528, 575, 601, 637, 677, 714, 736]$. Dĺžky dielikov na pravítku, teda rozdiely $37-5, 80-37, \dots, 736-714$, predstavujú dĺžky tónov. Dostávame vektor d : $[32, 43, 29, 51, 23, 44, 34, 30, 33, 48, 31, 38, 27, 41, 19, 47, 26, 36, 40, 37, 22]$.

Pozrime sa na praktickú stránku prepisu do nôt. Vektor v obsahuje čísla od 1 po 22, ktorým potrebujeme nájsť ekvivalent na klavíri. Ak by sme im priradili klávesy 1 až 22 tak ako to popisujeme v kapitole 1, šlo by o veľmi nízke tóny. Preto celú melódiu posunieme vyššie. Číslu 1 priradíme tón A v 2. oktáve, klávesu číslo $3 + 10 + 12 \times 2 = 37$. Urobíme teda posun $v \rightarrow v + 36$. Costasovo pole ostane po takejto transformácii Costasovým poľom, pretože posunutie o konštantu nenaruší ani jednu z jeho definičných vlastností v Def. 6.

Ďalej potrebujeme vyriešiť problém rôznych dĺžok vektorov v a d . Začali sme s $p = 23$, čo vedie k poľu 22×22 a vektoru v s 22 zložkami. Pole sme rozložili na pravítko dlhé 22, na ktorom je len 21 úsekov medzi značkami, preto má vektor d len 21 zložiek. Máme výšky pre 22 tónov a dĺžky len pre 21, preto vynecháme posledný tón. Melódia ostane bez opakovania výšok, pretože odobratím posledného tónu akurát zmenšíme množinu všetkých výškových rozdielov medzi dvojicami tónov. V pôvodnej množine boli všetky rôzne, budú aj v jej podmnožine.

Noty budeme zapisovať v programe MuseScore, ktorého najmenšia jednotka je 128-nová nota. Ostatné dĺžky nôt sú jej násobky:

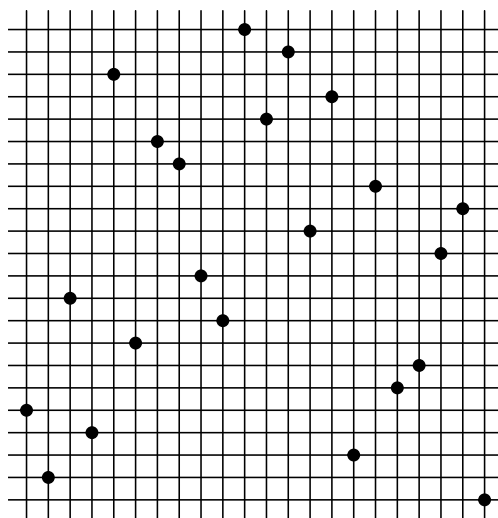
Nota	V násobkoch 128-novej
64-nová	$2 \times 128\text{-nová}$
32-nová	$4 \times 128\text{-nová}$
16-nová	$8 \times 128\text{-nová}$
osminová	$16 \times 128\text{-nová}$
štvrtová	$32 \times 128\text{-nová}$
polová	$64 \times 128\text{-nová}$
celá	$128 \times 128\text{-nová}$

Pre každý tón chceme vedieť, z koľkých nôt jednotlivých dĺžok sa skladá. Presne to nám povie zápis zložiek vektora d v dvojkovej sústave. Napríklad $d[1] = 32$ je v dvojkovej sústave 100000, takže prvý tón má byť $0 \times 128\text{-nová} + 0 \times 64\text{-nová} + 0 \times 32\text{-nová} + 0 \times 16\text{-nová} + 0 \times 8\text{-nová} + 1 \times \text{štvrtová}$ nota. Ďalej $d[2] = 43$, čo je v dvojkovej sústave 101011, takže druhý tón bude ligúra zo štvrtovej, 16-novej, 32-novej a 64-novej noty. Keďže žiadna zo zložiek vektora d nie je väčšia ako 2^6 , postačia nám 6-ciferné čísla v dvojkovej sústave, resp. najdlhšia nota v ligatúre bude štvrtová.

Tento prístup pretransformovania Golombovho pravítka je naším vlastným, preto sa môže v detailoch odlišovať od Rickardovho. Nie je známe, akým spôsobom ho vygeneroval - mohlo mať $m = 89$ značiek a teda sa na ňom nachádzali dĺžky pre všetkých 88 tónov v Rickardovej sonáte. Taktiež nie je známe, ako prerobil Golombovo pravítko na dĺžky tónov v notách a akú najkratšiu notu mal k dispozícii.

Výstupom programu *hudba.r* je súbor *melodiap.txt*, kde sú zapísané vektory v , jeho posunutá verzia zapísaná v tónoch, vektor d a vektor d v dvojkovej sústave (pozri prílohu B).

Pre lepšiu predstavu ilustrujeme Costasovo pole pre $p = 23$ a $\lambda = 5$ na Obr. 4.1

Obr. 4.1: Costasovo pole pre $p = 23$ a $\lambda = 5$

Na Obr. 4.2 uvádzame notový zápis vektorov v a d pre $p = 23$ a $\lambda = 5$. Túto melódiu nazveme Melódia Costas a v zvukovej forme ju uvádzame ako prílohu - pozri súbor *MelodiaCostas.mp3*.

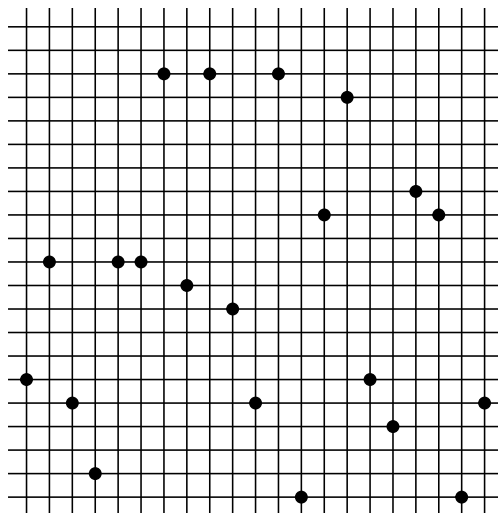
Obr. 4.2: Notový zápis pre $p = 23$ a $\lambda = 5$ (Melódia Costas)

Pre náš experiment ešte potrebujeme náhodne vygenerované melódie, ktoré generujeme programom *nahodne.r* (pozri prílohu A). Pre zadané n vytvoríme n -zložkový vektor v , zvolíme $n = 21$. Tóny budeme generovať z rovnomerného rozdelenia v rovna-

kom výškovom rozsahu ako v Melódii Costas, teda od A_2 po $F\sharp_4$. Dĺžky tónov zvolíme tiež rovnomerne náhodne v rozmedzí od najkratšieho po najdlhší vyskytujúci sa v Melódii Costas, teda od 22 po 48, resp. od 10110 po 110000. Ďalej zistíme, koľko sa v melódii nachádza rovnobežníkov a ekvidistančných trojíc na priamke pomocou funkcií $TROJICE(n, vysky, por).r$ a $ROVNOBEZ(n, vysky, por).r$ (pozri prílohu A). Tieto čísla hovoria, koľkokrát melódia porušuje definíciu Costasovho poľa, resp. melódie bez pravidelnosti. Ich súčet budeme chápať ako mieru pravidelnosti v melódii v zmysle opakovania výškových rozdielov.

V experimente budeme skúmať či je melódia bez pravidelnosti výšok a dĺžok tónov pre poslucháča výrazne horšia ako melódia s nejakou pravidelnosťou. Zároveň budeme testovať, či viac pravidelnosti môže implikovať vyššie hodnotenie od poslucháča. Za týmto účelom použijeme dve náhodne vygenerované melódie, kde jedna má takmer trikrát väčšiu mieru pravidelnosti výšok tónov ako druhá.

V prvej náhodnej melódii sa nachádza 9 rovnobežníkov a jedna ekvidistančná trojica na priamke. Na Obr. 4.3 ilustrujeme výšky zakreslené do mriežky.



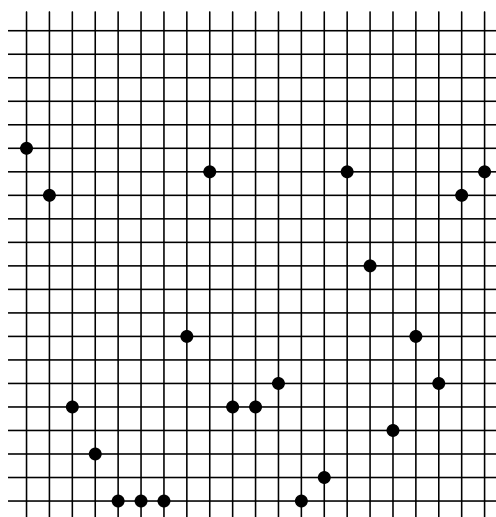
Obr. 4.3: Náhodne vygenerované výšky pre Melódiu 10 zakreslené do mriežky

Melódii ďalej vygenerujeme dĺžky tónov z rovnomerného rozdelenia v spomínanom rozsahu. Príslušný výstup programu *nahodne.r* je v prílohe B. Na Obr. 4.4 uvádzame prepis do nôt. Melódiu si môžeme vypočítať spustením súboru *Melodia10.mp3*.



Obr. 4.4: Notový zápis Melódie 10

V druhej náhodnej melódii je 22 rovnobežníkov a 6 ekvidistančných trojíc na priamke (pozri Obr. 4.5). Kompletný výstup programu *nahodne.r* aj s náhodnými dĺžkami sa nachádza v prílohe B. Na Obr. 4.6 uvádzame prepis do nôt. Samotná melódia je v súbore *Melodia28.mp3*.



Obr. 4.5: Náhodne vygenerované dĺžky pre Melódiu 28 zakreslené do mriežky

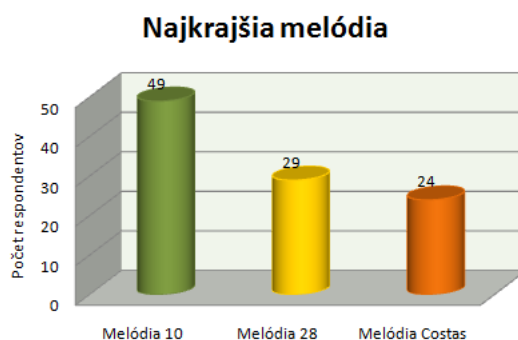


Obr. 4.6: Notový zápis Melódie 28

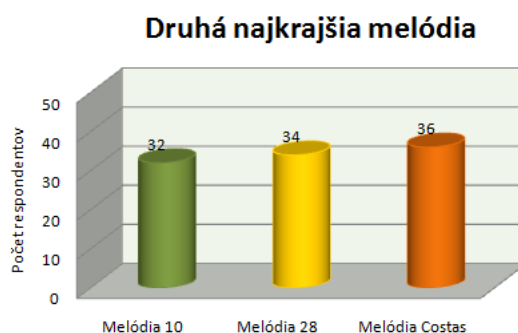
Použitím týchto troch melódií sme zostavili formulár pre experiment¹. Vo formulári sú označené nasledovne: Melódia 1 je Melódia 10, Melódia 2 je Melódia 28 a Melódia 3 je Melódia Costas. Formulár vyplnilo 102 respondentov, ktorých úlohou bolo zoradiť melódie podľa toho ako sa im páčia a následne im priradiť bodové ohodnotenie na škále 0-10 (čím vyššie tým krajšia). Pri zoradovaní bolo nutné zoradiť ich ostromonotónne, kým pri bodovom ohodnocovaní boli povolené dať viacerým melódiám rovnaké hodnotenie (urobili tak ôsmi opýtani). Respondenti boli požiadaní, aby melódiám, ktoré sa im páčia zhruba rovnako dali približne rovnaké hodnotenie a v opačnom prípade im dali hodnotenia s výraznejším rozdielom. Formulár ešte obsahoval otázku nakoľko je človek hudobníkom s tromi možnosťami: *hrám na nejaký hudobný nástroj*, *nehrám aktívne na žiadny hudobný nástroj* alebo *som profesionálny hudobník*. Kompletne formulácie otázok sú v prílohe C.

Pozrime sa na výsledky experimentu (kompletne dáta sú priložené na disku v súbore *data1.txt*). Prvý graf ukazuje koľko ľudí označilo príslušnú melódiu za najkrajšiu (Obr. 4.7). Vidíme, že vyhráva Melódia 10. Na ďalších dvoch grafoch vidíme, koľko ľudí označilo ktorú melódiu ako stredne peknú (Obr. 4.8) a najškaredšiu (Obr. 4.9). V kategórii najškaredšia tesne vedie Melódia Costas nad Melódiou 28.

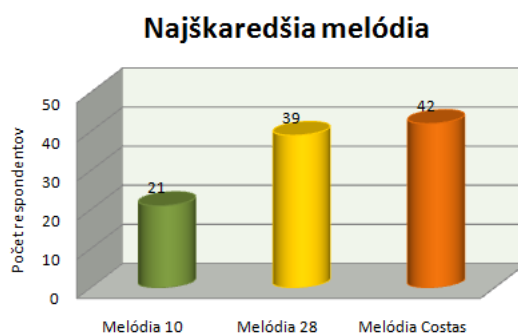
¹Dotazník je dostupný na <https://docs.google.com/forms/d/1-qEkjcL425C6ZLbWbXyKsDZ\SwrLeAZw3XUmPfiVSI0s/viewform>



Obr. 4.7: Graf zobrazujúci koľkí ľudia označili za najkrajšiu Melódiu 10, koľkí Melódiu 28 a koľkí Melódiu Costas

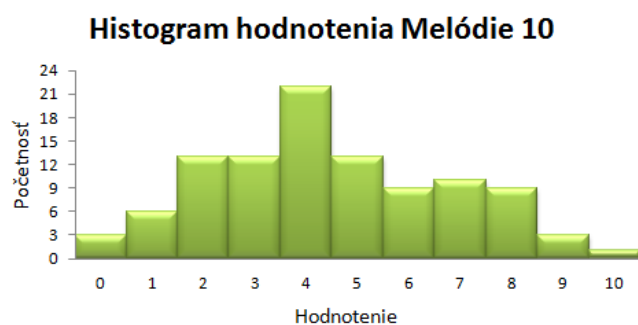


Obr. 4.8: Graf zobrazujúci koľkí ľudia označili za stredne peknú Melódiu 10, koľkí Melódiu 28 a koľkí Melódiu Costas



Obr. 4.9: Graf zobrazujúci koľkí ľudia označili za najškaredšiu Melódiu 10, koľkí Melódiu 28 a koľkí Melódiu Costas

Pozrime sa na bodové ohodnotenia. Na obrázkoch vidíme histrogramy číselného hodnotenia melódii. Výrazne najčastejšie hodnotenie Melódie 10 je 4.

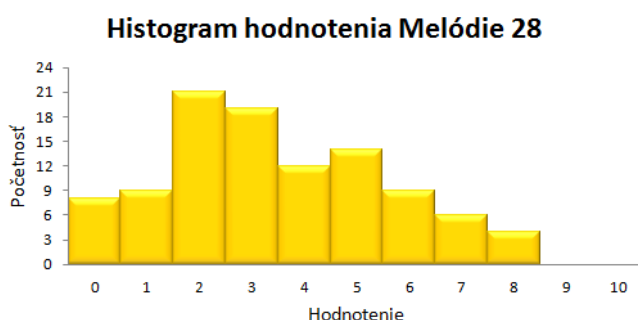


Obr. 4.10: Hodnotenia Melódie 10

Hodnotenie	Početnosť
0	3
1	6
2	13
3	13
4	22
5	13
6	9
7	10
8	9
9	3
10	1

Tabuľka 4.1: Číselné hodnotenia Melódie 10

Na Obr. 4.1 vidíme, že Melódia 28 dostávala najčastejšie hodnotenie 2 (21-krát) a veľmi často hodnotenie 3 (19-krát)

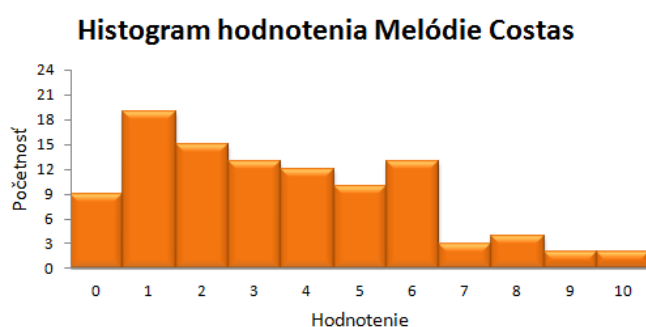


Obr. 4.11: Hodnotenia Melódie 28

Hodnotenie	Početnosť
0	8
1	9
2	21
3	19
4	12
5	14
6	9
7	6
8	4
9	0
10	0

Tabuľka 4.2: Číselné hodnotenia Melódie 28

V prípade Melódie Costas sa najčastejšie vyskytovalo hodnotenie 1. Je zaujímavé, že napriek tomu hodnotenie 6 dostala až 13-krát.



Obr. 4.12: Hodnotenia Melódie Costas

Hodnotenie	Početnosť
0	9
1	19
2	15
3	13
4	12
5	10
6	13
7	3
8	4
9	2
10	2

Tabuľka 4.3: Číselné hodnotenia Melódie Costas

Porovnajme aritmetické priemery bodových hodnotení. Melódia 28 a Melódia Costas majú takmer rovnaké priemerné hodnotenie, Melódia 10 približne o 1 väčšie. Aj tu sa ukazuje, že hoci je rozdiel v miere pravidelnosti medzi melódiami 28 a Costas najväčší, respondenti ich hodnotia ako približne rovnako škaredé.

Melódia 10	Melódia 28	Melódia Costas
4,48	3,47	3,48

Tabuľka 4.4: Aritmetický priemer bodových hodnotení.

Z výsledkov experimentu nadobúdame tušenie, že celkovo je najkrajšia Melódia 10 a najškaredšia Melódia Costas alebo Melódia 28. Toto tušenie štatisticky overíme porovnaním stredných hodnôt bodových hodnotení (ďalej len stredná hodnota melódie).

Označme strednú hodnotu Melódie Costas μ_C , strednú hodnotu Melódie 10 μ_{10} a strednú hodnotu Melódie 28 μ_{28} . Prvým cieľom je potvrdiť, alebo vyvrátiť tvrdenie profesora Rickarda, že melódia bez pravidelnosti je ďaleko najškaredšia. Druhým cieľom je zistiť, či je melódia s väčšou mierou pravidelnosti v priemere vyššie hodnotená ako melódia s menšou. Budeme testovať takúto hypotézu:

Hypotéza 2. $H_0 : \mu_C = \mu_{10} = \mu_{28}$ vs. $H_1 : \text{aspoň jedna rovnosť nie je splnená.}$

Pre porovnávanie viacerých stredných hodnôt súčasne bez straty 95% spoľahlivosti treba podľa [8] použiť tzv. ANOVA (Analysis of Variance). Táto metóda vyžaduje normalitu, nezávislosť dát a tiež rovnosť disperzií. Normalita bola pre všetky melódie zamietnutá Shapirovym-Wilkovým testom (pozri [12]) a dáta sú prirodzene závislé, keďže všetky tri melódie hodnotil ten istý človek. Preto použijeme verziu metódy ANOVA, ktorá nevyžaduje spomínané predpoklady.

Máme situáciu, kedy sme jeden pokus opakovali viac krát za rôznych podmienok (dali sme počúvať tie isté melódie rôznym ľuďom) a chceme porovnať viacero stredných hodnôt súčasne. Podľa [8] treba použiť tzv. Repeated Measures ANOVA (RM ANOVA). Tento test je rozšírením Studentovho párového dvojvýberového t-testu pre 3 a viac objektov. Tieto objekty, u nás Melódia 10, Melódia 28 a Melódia Costas, budeme nazývať *skupiny*². Od každého zo 102 respondentov máme hodnotenie vo všetkých

²z angl. groups

troch skupinách. Trojice hodnotení od jednotlivých ľudí budeme volať *bloky*³. Máme teda maticu X rozmerov 102×3 a číslami rozsahu 0-10, kde stĺpce predstavujú skupiny a riadky bloky.

Metóda RM ANOVA umožňuje testovať dáta, kde sú hodnoty v rámci blokov závislé. Stále však vyžaduje normalitu. Preto použijeme neparametrickú verziu RM ANOVA, Quadeho test, ktorý funguje aj pre nenormálne dáta s rôznymi disperziami. Budeme postupovať podľa [8]. Základnou ideou testu je naváhovať výchylky čísel od blokového priemeru podľa rozptylu celého bloku. Keďže ide o neparametrickú verziu, použijeme *ranky* (zoradenie podľa veľkosti). Bloky očísľujeme číslami $Q_1, Q_2, \dots, Q_{102}$ podľa maximálneho rozdielu čísel v bloku. Ďalej očísľujeme trojice hodnôt v rámci blokov číslami $R(X_{ij}) \in \{1, 2, 3\}$. Číslo bloku Q_i potom vynásobíme rozdielom vnútroblokového $R(X_{ij})$ a priemeru, čím dostaneme čísla S_{ij} :

$$S_{ij} = Q_i \left(R(X_{ij}) - \frac{3+1}{2} \right).$$

Sčítaním po stĺpcoch dostaneme účet pre každú skupinu:

$$S_j = \sum_{i=1}^{102} S_{ij}^2.$$

Ďalej vypočítame celkový súčet štvorcov (*total sum of squares* - TSS) a súčtu štvorcov pre skupiny (*treatment sum of squares* - tSS):

$$TSS = \sum_{i=1}^{102} \sum_{j=1}^3 S_{ij}^2, \quad tSS = \frac{1}{102} \sum_{j=1}^3 S_j^2$$

Na záver vypočítame testovaciu štatistiku,

$$T_Q = \frac{(102-1)tSS}{TSS - tss} \doteq 0.020374,$$

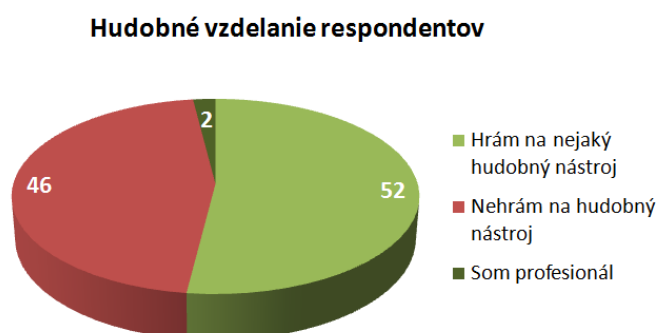
ktorú porovnáme s 5%-nou kritickou hodnotou Fisherovho-Snedecorovho F-rozdelenia s príslušnými stupňami voľnosti: $F_{101,2} \doteq 19.4858$. Vidíme, že T_Q sa nenachádza za kritickou hodnotou. Ekvivalentne o tom hovorí aj p-hodnota = 97.98%, čo je ďaleko viac ako 5%, teda H_0 nezamietame (pozri kód *experiment.r*, časť Quadeho test). Tento výsledok hovorí, že skutočné stredné hodnoty sú približne rovnaké, nemožno ich považovať za rôzne. Prišli sme k záveru, že na základne nášho experimentu nezníe melódia vytvorená pomocou Costasovho poľa a Golombovho pravítka výrazne horšie než náhodne vygenerovaná melódia, ako tvrdí profesor Rickard v [9].

³z angl. blocks

Zdá sa, že naše ucho nepočuje signifikantný rozdiel medzi melódiou bez pravidelnosti a náhodne vygenerovanou, v ktorej sa trochu pravidelnosti nachádza. Tento výsledok treba brať s rezervou, pretože ide o konkrétny tip náhodnej melódie (generovanej z rovnomerného rozdelenia). Taktiež by experiment mohol dopadnúť inak ak by sme použili melódie inej dĺžky. Je možné, že naše ucho potrebuje dlhšiu melódiu, aby stihlo zaregistrovať absenciu pravidelnosti alebo postrehnúť rozdiel v jej miere. Porovnanie dlhších náhodných a klasických melódií z hľadiska pravidelnosti je v kapitole 4.3.

4.2 Súvislosť s hudobným vzdelaním

V tejto časti otestujeme, či výber najškaredšej melódie závisí od hudobnej kategórie respondenta. Na Obr. 4.13 vidíme rozdelenie do kategórií. Dvaja ľudia neoznačili do ktorej kategórie patria, preto v tejto časti pracujeme zo 100 respondentmi.



Obr. 4.13: Počty respondentov, ktorí sa zaradili do jednotlivých kategórií

Keďže do kategórie Profesionál sa zaradili len 2 ľudia, zlúčime ju s kategóriou Hráť. Pozrime sa ako v priemere hlasovali ľudia z rovnakej kategórie.

	Melódia 10	Melódia 28	Melódia Costas
Hráť	4,87	3,61	3,33
Nehráť	3,91	3,26	3,59

Tabuľka 4.5: Priemerné bodové hodnotenia melódií v podľa hudobných kategórií.

V oboch kategóriách vyhrala Melódia 10. V prípade najhoršie hodnotenej melódie sa to rôzni: v kategórii Hráť dostala v priemere najnižšie hodnotenie Melódiu Costas, kým v kategórii Nehráť sa zdá byť najškaredšia Melódia 28.

Zavedme dve náhodné veličiny M (melódia) a H (hudobná kategória), ktoré charakterizujú výber najškaredšej melódie v prípade hudobníkov a nehudobníkov. Veličina M nadobúda hodnoty 10, 28 a 235, ktoré prislúchajú výberu Melódie 10, Melódie 28 a Melódie Costas (s parametrami $p = 23, \lambda = 5$) a veličina H nadobúda hodnoty ' $Hram$ ', ' $Nehram$ '. Urobíme χ^2 test o nezávislosti H a M na 5%-nej hladine spoľahlivosti podľa [16, s. 195-196]. Testujeme hypotézu:

Hypotéza 3. $H_0 : M$ a H nie sú nezávislé vs. $H_1 : M$ a H sú nezávislé.

K tomuto testu potrebujeme napočítať nasledovné početnosti:

$$P_{ij} = \text{početnosť javu } [M = i, H = j] \text{ pre } i = 1, 2, 3, j = 1, 2$$

t.j. napr. pre $i=1, j=2$ je P_{ij} počet koľko krát vybrali nehudobníci za najškaredšiu Melódiu 10. V tabuľke 4.6 sú P_{ij} čísla v obdĺžniku 2.-3. stĺpec a 2.-5. riadok. Ďalej

$$P_{i.} = \sum_{j=1}^2 P_{ij}, \quad P_{.j} = \sum_{i=1}^3 P_{ij}$$

kde suma vľavo predstavuje celkový počet ľudí, ktorý vybrali melódiu i . V tabuľke 4.6 tomu zodpovedá stĺpec 4, riadky 2-4. Suma vpravo predstavuje celkový počet ľudí, ktorý sa označili za hudobníka, resp. nehudobníka. V tabuľke 4.6 tomu zodpovedá riadok 5, stĺpce 2-3.

	H=Hram	H=Nehram	Spolu
M=10	7	14	21
M=28	20	17	37
M=235	27	15	42
Spolu	54	46	100

Tabuľka 4.6: Početnosti výberu najškaredšej melódie hudobníkmi a nehudobníkmi.

Testovaciu štatistiku X^2 vypočítame ako

$$X^2 = \sum_{i=1}^3 \sum_{j=1}^2 \frac{\left(P_{ij} - \frac{P_{i.}P_{.j}}{n}\right)^2}{\frac{P_{i.}P_{.j}}{n}}$$

kde $n = 100$ respondentov. Za platnosti H_0 má X^2 asymptoticky rozdelenie χ^2 s počtami stupňov voľnosti $(3 - 1)(2 - 1) = 2$. Hodnotu X^2 vypočítame programom

experiment.r, časť χ^2 test (pozri prílohu A) a porovnáme ju s 5%-nou kritickou hodnotou $\chi^2_2(5\%) = 5.9915$. Hodnota štatistiky vyjde $X^2 = 5.3997$, čo je menej ako $\chi^2_2(5\%)$, takže H_0 nezamietame. Ekvivalentne o tom hovorí p -hodnota $= 6.722\% > 5\%$.

Napriek tomu, že v experimente vyhrala v kategórii najškaredšia u hudobníkov Melódia Costas a u nehudobníkov Melódia 28, test povedal, že to nie je dostatočne štatisticky významné na to, aby sme mohli prehlásiť preferencie hudobníkov a nehudobníkov za odlišné. Záverom testu je, že výber najškaredšej melódie zrejme nezávisí od toho, či je človek hudobník alebo nie.

4.3 Miera pravidelnosti v náhodnej a klasickej melódii

Je možné, že v náhodných melódiách bolo primálo pravidelnosti na to, aby to naše ucho dokázalo rozlíšiť od melódie bez pravidelnosti. Porovnajme ich pravidelnosť s niekoľkými klasickými skladbami.

Melódie budeme porovnávať len z hľadiska pravidelnosti výšok tónov (ďalej len miera pravidelnosti) a to ako súčet počtu rovnobežníkov a ekvidistančných trojíc na priamkach. Pravidelnosť dĺžok tónov nebudeme uvažovať z dvoch dôvodov: po prvé, je ťažké vôbec rozoznať dĺžky tónov vo vygenerovaných melódiách, pretože sú podobné. Pri tempe $\text{♩}=120$ trvá štvrtková nota len $\frac{60}{120} = \frac{1}{2}$ sekundy a najkratšia 128-nová nota len $\frac{1}{2^5} \frac{1}{2} = 0.015625$ sekundy; po druhé, mieru pravidelnosti v melódiách z experimentu budeme porovnávať s melódiou, ktoré na rozdiel od našich a Rickardových obsahuje pomlčky. Pri našom koncepte dĺžok tónov ako pravítka pomlčky neuvažujeme.

Najprv porovnáme klasickú melódiu s melódiami z experimentu. Klasickú skladbu prepíšeme v programe MuseScore na zjednodušenú monofonickú melódiu. Zvolíme skladbu Pre Elišku už spomínaná v kap. 2, kde sme použili prvú 17-tónovú ucelenú časť. Melódie z experimentu majú 21 tónov, preto sa potrebujeme pozrieť na rozsiahlejšiu časť melódie Pre Elišu. Pozrime sa ako pokračuje ďalej (Obr. 4.14, pozri [1]), melódia je aj v súbore *PreElisku35.mp3*.

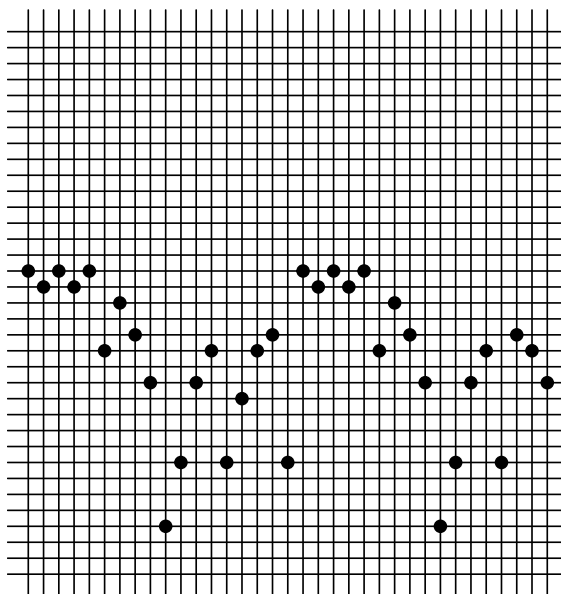
Pre Elišku

Ludwig van Beethoven

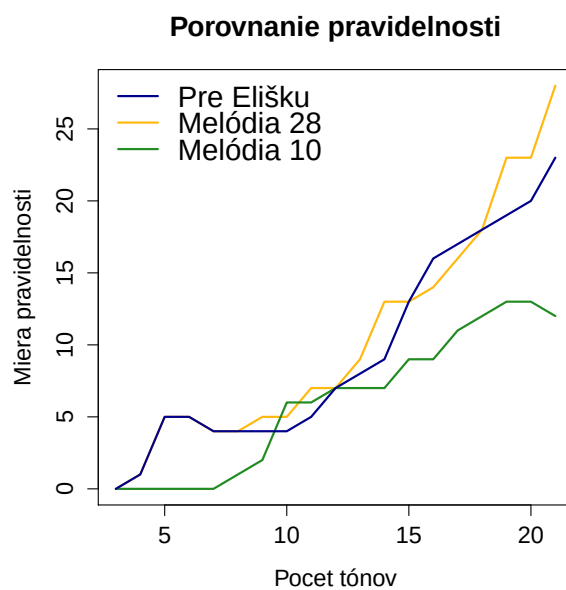


Obr. 4.14: Notový zápis úryvku zo skladby Pre Elišku.

Zápis vo vektoroch v a d získame pomocou programu *pravidelnost.r* (pozri prílohu A). Vykreslenú melódiu v mriežke ilustrujeme na Obr. 4.15. Pozrime sa pomocou programu *pravidelnost.r* na rast pravidelnosti s rastúcim počtom tónov vo všetkých troch melódiách. Pre každú začneme od troch tónov s každým pridaným tónom spočítame mieru pravidelnosti pomocou funkcií $TROJICE(n, vysky, por).r$ a $ROVNOBEZ(n, vysky, por).r$. Výsledok vidíme na Obr. 4.16. Pre 21 tónov majú Melódia 28, Pre Elišku a Melódia 10 mieru pravidelnosti 28, 23 a 10. Možeme konštatovať, že pri takejto dlhej melódii je miera pravidelnosti 23 dostačujúca na to, aby sa melódia dala považovať za peknú, dokonca bola úvodom hlavnej melódie jednej z najznámejších klasických skladieb (podľa [4]). Melódia 28 teda mala z tohoto hľadiska šancu na úspech. Napriek tomu pri výbere najkrajšej melódie preferovalo až o $\frac{1}{5}$ opýtaných viac Melódiu 10 oproti Melódii 28.



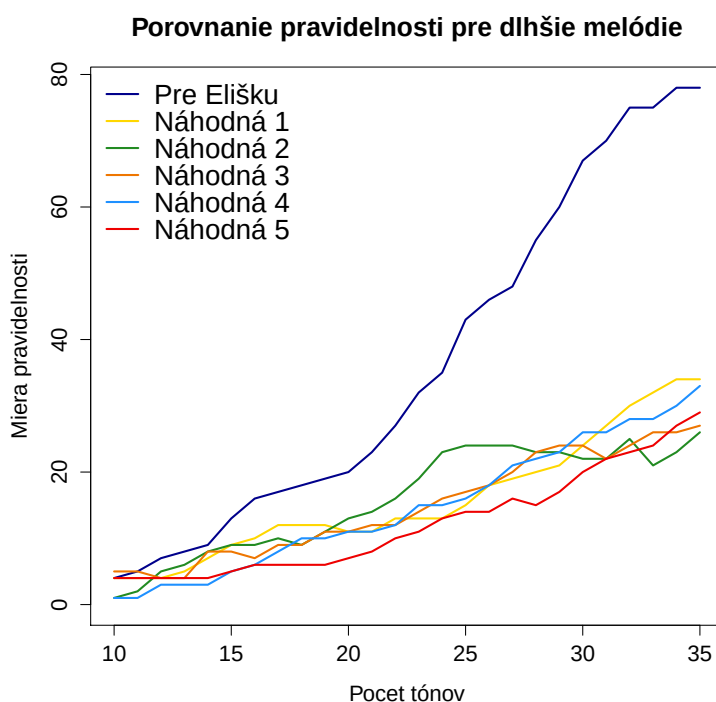
Obr. 4.15: Prepis melódie Pre Elišku 21 do mriežky programom *pravidelnost.r*



Obr. 4.16: Závislosť miery pravidelnosti od počtu tónov melódií Pre Elišku, Melódia 28 a Melódia 10

Hoci melódia Pre Elišku nebola najpravidelnejšia z melódií v experimente, vo vše-

obecnosti zrejme bude ďaleko pravidelnejšia ako náhodné melódie, hlavne pri porovnávaní dlhších úsekov. Porovnajme ju s ďalšími náhodnými melódiami väčšej dĺžky. Z rovnomerného rozdelenia vygenerujeme melódie dĺžky 35 a porovnáme vývoj pravidelnosti od 10 po 35 tónov. Na Obr. 4.17 vidíme, že pre dlšie melódie je rozdiel v pravidelnosti výraznejší - pri Eliške je miera pravidelnosti pre 35 tónov 78, pre náhodné melódie je to okolo 25.



Obr. 4.17: Porovnanie pravidelnosti melódií Pre Elišku a piatich náhodných

Melódia Pre Elišku sa zdá byť dosť pravidelná. Urobme ešte jedno porovnanie náhodných melódií s klasickou skladbou, ktorá znie „divokejšie“. Vyberieme skladbu *Nocturne B♭ minor, Op. 9 No.1* od Frederica Chopina a použijeme jej začiatok, kde sa nenachádzajú pomlčky. Prvých 35 tónov sme prepísali v programe MuseScore, pozri Obr. 4.18, [5]. Melódia je v súbore *Nocturne35.mp3*.

Porovnanie s ďalšími náhodnými melódiami je na Obr. 4.19. Aj táto skladba má omnoho viac pravidelnosti ako náhodné melódie.

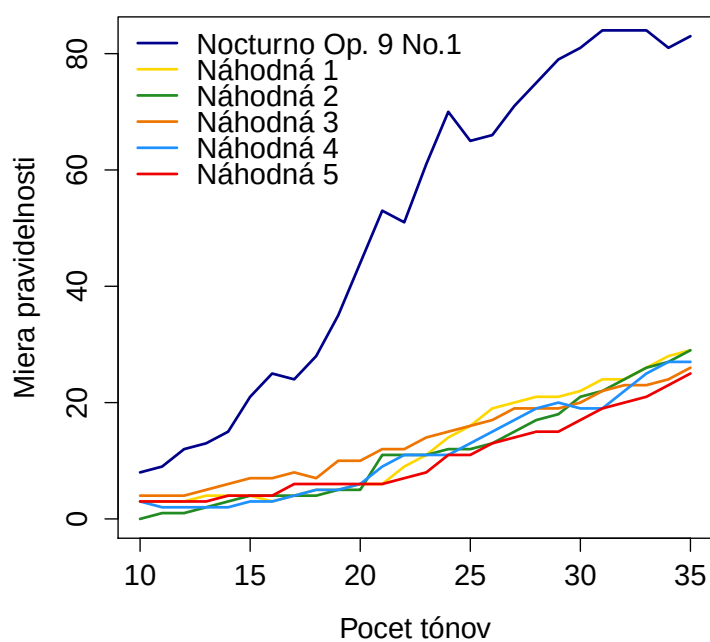
Nocturne in B $\flat$ minor, Op. 9 No.1

Frederic Chopin



Obr. 4.18: Notový zápis úryvku zo skladby Nocturno B $\flat$ Op.9 No.1

Porovnanie pravidelnosti pre dlhšie melódie



Obr. 4.19: Porovnanie pravidelnosti melódií Pre Elišku a piatich náhodných

Záver

V práci sme najprv zaviedli prepis melódie do vektorov v a d pomocou kláves na klavíri. Rozobrali sme matematiku za prednáškou prof. Rickarda využitím pojmov Golombovo pravíko, Costasovo pole, primitívny koreň mod n . Pomocou Malej Fermatovej vety sme vo Vete o generovaní 3 ukázali, že Costasovo pole možno pre prvočíslo p skonštruovať umocňovaním primitívneho koreňa λ a modulovaním p . Popísali sme spôsob rozloženia Costasoho poľa na Golombovo pravítko podľa [2]. Vytvorili sme programy *primkoren.wmx*, *costas.r* a funkciu *COSTAS_GOLOMB($n, vysky$)*, pomocou ktorých sa dajú generovať Golombove pravítka a Costasove polia a to aj veľkosti tretieho rádu.

Popísali sme spôsob, akým Rickard vytvoril sonátu a zreplikovaním jeho postupu sme vytvorili melódiu bez pravidelnosti. Náš postup sa môže v detailoch líšiť od Rickardovho, ako to popisujeme v kapitole 4.1.

Vygenerovali sme dve melódie z rovnomerného rozdelenia a spolu s melódiou bez pravidelnosti sme zrealizovali experiment pre ich porovnanie z hľadiska poslucháča. Na vzorke 102 respondentov sme v časti 4.1 Quadeho testom nezamietli hypotézu, že stredné hodnoty hodnotení melódií sa líšia. Výsledkom testu je vyvrátenie tvrdenia prof. Rickarda, že melódia bez pravidelnosti znie ďaleko horšie ako náhodne vygenerovaná melódia. Tento výsledok sa týka generovania z rovnomerného rozdelenia a melódie dĺžky 21 tónov.

V časti 4.2 sme použitím χ^2 testu nezamietli hypotézu, že výber najškaredšej melódie nezávisí od hudobnej kategórie poslucháča, aj keď podľa výsledkov experimentu by sa tak na prvý pohľad mohlo zdať.

Vytvorili sme R-kové funkcie *ROVNOBEZ($n, vysky, por$)* a *PRIAMKY($n, vysky, por$)* pre určenie miery pravidelnosti melódie. Pomocou nich sme porovnali pravidelnosť

v melódii Pre Elišku s náhodných melódiách z experimentu. Zistili sme, že náhodná Melódia 28 obsahuje dokonca mierne viac pravidelnosti ako Pre Elišku ak uvažujeme 21 tónov. V tejto melódii teda bolo dostatočne veľa pravidelnosti aby sa mohla páčiť poslucháčovi. Napriek tomu bola skupinou nehudobníkov označená za najškaredšiu a vyhrala Melódia 10 s podstatne menšou mierou pravidelnosti. Rozdiel v hodnoteniach sa však neukázal byť štatisticky významný. Záverom je, že miera pravidelnosti ako počet porušení definície Costasovho poľa 6 skôr nesúvisí s tým, ako sa melódia páči poslucháčovi. Špeciálne 0 porušení, teda melódia bez pravidelnosti sa neukázala byť naškaredšia.

Na záver chcem poznamenať, že cieľom práce nebolo komponovať skladby pomocou algoritmov a teoretickej matematiky, ale ich analyzovať. Komponovanie hudby patrí v mojich očiach výlučne kreativite ľudí.

I used to look at composing
music as problem solving. But
as I get older, it's not about
problem solving anymore. There
are no solutions, because there
are no problems. You just turn
the tap and it flows out.[15]

John Zorn

Literatúra

- [1] BEETHOVEN, Ludwig: *Pre Elišku*, Slovenský hudobný fond, Bratislava, 1991
- [2] K. Drakakis, S. Rickard: *On the Construction of Nearly Optimal Golomb Rulers by Unwrapping Costas Arrays*, dostupné na internete (10.11.2015):
<http://www.m-hikari.com/ces/ces2010/ces5-8-2010/drakakiskCES5-8-2010.pdf>
- [3] DROPPOVÁ, Antónia: *Elementárna hudobná teória*, Učebný text pre odbor učiteľstvo pre I. st. ZŠ, Prešov, 1998, dostupné na internete (29.3.2016):
http://hudobnik.wbl.sk/hudobna_nauka/elementarna_hudobna_teoria.pdf
- [4] FULD, James J.: *The Book of World-famous Music: Classical, Popular, and Folk*.
- [5] CHOPIN, Frederic: *Nocturnes Vol. V*, Schlesinger'sche Buch-und Musikhandlung, Berlin, 1881
- [6] IBM: *Table of lengths of shortest known Golomb rulers*, dostupné na internete (10.11.2015):
<http://www.research.ibm.com/people/s/shearer/grtab.html>
- [7] KLEINOVÁ, Katarína: *pedagogička a violončelistka orchestra Slovenského rozhlasu, osobná komunikácia, Základná umelecká škola Ludovíta Reitera*, Bratislava, 2016
- [8] OFUNGWU, Joseph: *Statistical Applications for Environmental Analysis and Risk Assessment*, časť 9.4.4.
- [9] RICKARD, Scott: *The beautiful math behind the world's ugliest music*, video, Konferencia TEDxMIA, dostupné na internete (15.9.2015):

https://www.ted.com/talks/scott_rickard_the_beautiful_math_behind_the_ugliest_music

- [10] RIBENBOIM, Paulo: *The Book of Prime Number Records*, Springer Science & Business Media, 6. 12. 2012, 479 s.
- [11] SCHMUCKLER, Mark A.: *Expectations in music: Investigation of Melodic and Harmonic processes*, Music Perception Vol. 7 No. 2, 109-150, 1989, California, dostupné na internete (1.12.2015):
<http://utsc.utoronto.ca/~marksch/Schmuckler%201989.pdf>
- [12] SOMORČÍK, Ján: Počítačová štatistika, kurzy pre EFM, zimný semester 2016
- [13] TIMM, Matthew: *Math in the Middle Institute Partnership Expository Paper*, dostupné na internete (30.10.2015):
http://scimath.unl.edu/MIM/files/MATExamFiles/TimmMatt_070511_Final-LA.pdf
- [14] TEDxMIA: O Prof. Rickardovi, dostupné na internete (10.10.2015):
https://www.ted.com/speakers/scott_rickard
- [15] Brainy Quote, webová stránka, dostupné na internete (1.5.2016):
<http://www.brainyquote.com/quotes/keywords/composing.html>
- [16] Zvára, K., Štěpán, J.: *Pravděpodobnost a matematická statistika*, Matfyzpress, Praha, 1997

Príloha A

V tejto prílohe sú zdrojové kódy k programom spomínaným v texte.

Spustiteľné verzie sa nachádzajú na disku.

A.1 Zdrojové kódy v jazyku Maxima

Program *primkoren.wxmx*

```
1 p: 11;
2 ciska: matrix()$
3 hranica: min(p,20)$
4 for a: 2 thru hranica do ciska: addrow(ciska,[a])$
5 pocet: length(ciska)$
6 zvysky: matrix()$
7 for i: 1 thru pocet do
8   (zvysky: matrix(),
9    aktual: ciska[i],
10    for j: 1 thru (p-1) do (dalsi: mod(aktual^j,p),
11     zvysky: addrow(zvysky,[dalsi])),
12    if (length(unique(args(zvysky))) = (p-1)) then
13     return(aktual))$
14 aktual;
15 float(zvysky);
16 write_data(zvysky,"zvysky.txt");
```

A.2 Zdrojové kódy v jazyku R

Program *mriezky.r*, ktorý slúži na vykresľovanie rôznych mriežok.

```
1 #pred vykreslením vygeneruj vysky (a por) jedným so 4 spôsobov:
2
3 n <- 7
4 vysky <- c(4,-1,1,-1,6,-1,3)
5
6 #obr. 2.2a
7 n <- 9
8 vysky <- c(7,-1,6,-1,-1,-1,4,-1,3)
9
10 #obr. 2.2b
11 n <- 7
12 vysky <- c(3,-1,-1,4,-1,-1,5)
13
```



```

14 #obr. 2.4
15 n <- 7
16 vysky <- c(6,9,7,1,3,4,9,2,5,3)
17
18 ##2## nahodne z rovnomerneho rozdelenia
19 n <- 10
20 vysky <- sample(1:n, n, replace=TRUE)
21
22 ##3## Costasovo pole malych rozmerov
23 #Obr.2.3
24 lambda <- 2          #primitive root mod ohr
25 p <- 11              #prvocislo
26 n <- p-1
27 vysky <- mat.or.vec(n,1)
28 for (i in 1:ohr-1) vysky[i] <- lambda^i %% p      ## je modulo
29
30 #Obr. 3.2
31 lambda <- 3          #primitive root mod ohr
32 p <- 7              #prvocislo
33 n <- p-1
34 vysky <- mat.or.vec(n,1)
35 for (i in 1:ohr-1) vysky[i] <- lambda^i %% p
36
37 ##4## Costasovo pole vacsich rozmerov, treba generovat v Maxime
38 #Obr. 3.1, p=19
39 vysky <- read.table("c:\\maxima-5.37.1\\wxMaxima\\zvysky.txt",
40 header=FALSE, sep=' ', )
41 vysky <- as.matrix(vysky)[,1]
42
43 #Obr. 4.1, p=23
44 vysky <- read.table("c:\\maxima-5.37.1\\wxMaxima\\zvysky.txt",
45 header=FALSE, sep=' ', )
46 vysky <- as.matrix(vysky)[,1]
47
48 ##VYKRESLENIE##
49 KRESLI(vysky)

```

Program *costas.r*

```

1  ohr <- 11          #prvocislo z Maximy
2  n <- ohr-1        #rozmer mriezky
3
4  #lambda <- 2      #pritivny koren mod ohr
5  #vysky <- mat.or.vec(n,1)
6  #for (i in 1:ohr-1) vysky[i] <- p^i %% ohr      ## je modulo
7
8  #nacistanie y-sur. Costasovho pola z Maximy
9  vysky <- read.table("c:\\maxima-5.37.1\\wxMaxima\\zvysky.txt",
10 header=FALSE, sep=' ', )
11 vysky <- as.matrix(vysky)[,1]
12
13 por <- c(1:n)
14 hudba <- cbind(por, vysky)
15
16 #vykreslenie
17 KRESLI(vysky)
18
19 #hľadanie rovnobezníkov a trojíc bodov na priamke (ma vyjst 0 a 0)
20 #priamky
21 TROJICE(n, vysky, por)
22
23 #rovnobezníky
24 ROVNOBEZ(n, vysky, por)

```

Program *nahodne.r*

```

1  n <- 21
2
3  #generovanie s rovnomerneho rozdelenia – zakomentuj ak nacistavas
4  #z txt
5  vysky <- sample(1:n, n, replace=TRUE)
6  vzdial <- sample(22:48, n, replace=TRUE)
7  #konstanty podľa Golomba p=23, lambda=5
8  library(R. utils)
9  dlzkyv2 <- intToBin(vzdial)
10
11 #alebo nacistanie uz existujucich dat – zakomentuj ak generujes
12 #nacistane <- read.table("D:/melodia21_28.txt", header=TRUE, sep=' ')
13
14 #attach(nacistane)
15 #dlzkyv2 <- nacistane[,3]
16
17 por <- c(1:n)
18
19 #vykreslenie do mriezky
20 KRESLI(vysky)
21
22 #hľadanie rovnobežníkov a trojíc bodov na priamke:
23 #priamky
24 TROJICE(n, vysky, por)
25
26 #rovnobežníky
27 ROVNOBEZ(n, vysky, por)
28
29 stupnica <- c('a2', 'be2', 'h2', 'c3', 'cis3', 'd3', 'dis3', 'e3', 'f3',
30 'fis3', 'g3', 'gis3', 'a3', 'be3', 'h3', 'c4', 'cis4', 'd4', 'dis4', 'e4',
31 'f4', 'fis4')
32 tony <- rep(0, n)
33 for (i in 1:n) tony[i] <- stupnica[vysky[i]]
34
35 melodia <- data.frame(vysky, tony, dlzkyv2)
36 write.table(melodia, "D:/melodia21_28.txt")

```

Program hudba.r

```

1  ohr <- 23          #prvocislo
2  n <- ohr-1        #rozmer mriezky
3
4  vysky <- read.table("c:\\maxima-5.37.1\\wxMaxima\\zvysky.txt",
5  header=FALSE, sep='␣')
6  vysky <- as.matrix(vysky)[,1]
7
8  por <- c(1:n)
9
10 #vykreslenie
11 KRESLI(vysky)
12
13 #Costas -> Golomb. pravitko
14 gol_prav <- COSTAS_GOLOMB(n, vysky)
15
16 vzdial <- diff(golomb)          #dlzky not v 128novych notach
17 #dlzka not – zakl. jednotka 128–nova nota
18 #noty su v jednotkach 128nova, 64tinova, 32tinova, 16tinova, 8nova,
19 #stvrtova, polova
20 #pocet jednotiek -> zoznam vzdial v dvojrovej sustave
21 #101011 = stvrtova + 16tinova + 64tinova + 128minova
22
23 library(R. utils)
24 dlzkyv2 <- c(intToBin(vzdial), 0)
25
26 #vektor vysky na hudobne znacenie

```

```

27 stupnica <- c('a2','be2','h2','c3','cis3','d3','dis3','e3','f3',
28 'fis3','g3','gis3','a3','be3','h3','c4','cis4','d4','dis4','e4',
29 'f4','fis4')
30 tony <- rep(0,n)
31 for (i in 1:n) tony[i] <- stupnica[vysky[i]]
32
33 melodia <- data.frame(vysky,tony,golomb,dlzkyv2)
34 write.table(melodia, "D:/melodia23.txt")

```

Program *experiment.r*

```

1 data <- read.table("D:/data1.txt",header=TRUE, sep='\t')
2 attach(data)
3
4 #odhady strednej hodnoty
5 mean(Melodia10)
6 mean(Melodia28)
7 mean(MelodiaCostas)
8
9 #testy normality
10 shapiro.test(Melodia10)
11 #nie su z N
12
13 shapiro.test(Melodia28)
14 #nie su z N
15
16 shapiro.test(MelodiaCostas)
17 #nie su z N
18
19 #Quadeho test
20 Y <- matrix(c(data[,5],data[,6],data[,7]),
21             nrow = dim(data)[1], byrow = TRUE,
22             dimnames = list(Respondent = as.character(1:102),
23                             Melodia = c('Mel10','Mel28','Costas')))
24 quade.test(Y)
25 #p-value=97,98% > 5%, H0 nezamietame
26
27 #Chi-kvadrat test
28 #H0: su zavisle vs H1: su nezavisle
29 #H - Hudobnik, N - Nehudobnik
30 #ocistenie od chybajucich dat zlucenie Profesional=Hram
31 data[Hudobnik=='',8] <- NA
32 data[Hudobnik='Profesional',8] <- 'Hram'
33 data1 <- na.omit(data)
34 Hudobnik <- data1[,8]
35
36 #tabulka pocetnosti oznaceni za najskarediu
37 library(MASS)
38 tab <- data.frame((table(data1$Najmenej,data1$Hudobnik)
39 [,1:3]))[, -1])
40 tab
41
42 chisq.test(tab)
43 #p-value=6,72% > 5%, zavislost nezamietame

```

Program *pravidelnost.r*

```

1 stupnica <- c('a2','be2','h2','c3','cis3','d3','dis3','e3','f3',
2 'fis3','g3','gis3','a3','be3','h3','c4','cis4','d4','dis4','e4',
3 'f4','fis4','g4','gis4','a4','be4','c5','cis5')
4
5 #35 tonov
6 EliskaTony <- c('e4','dis4','e4','dis4','e4','h3','d4','c4','a3',
7 'c3','e3','a3','h3','e3','gis3','h3','c4','e3','e4','dis4','e4',
8 'dis4','e4','h3','d4','c4','a3','c3','e3','a3','h3','e3','c4',

```

```

9 | 'h3', 'a3')
10 | n <- length(EliskaTony)
11 | por <- c(1:n)
12 | Eliska <- rep(0,n)
13 | for (i in 1:length(EliskaTony)) Eliska[i] <-
14 | which(stupnica==EliskaTony[i])
15 |
16 | KRESLI(Eliska)
17 |
18 | #####porovnanie Elisky, Mel 28 a Mel 10, od 10 po 21 tonov#####
19 | od <- 3
20 | po <- 21
21 | n <- po
22 | PocTonov <- c(od:po)
23 |
24 | #matica melodii
25 | k <- 2 #k je pocet melodii okrem Elisky
26 | mel28 <- read.table("d:/melodia21_28.txt", header=TRUE)
27 | mel10 <- read.table("d:/melodia21_10.txt", header=TRUE)
28 | melodie21 <- cbind(Eliska[1:n], mel28[,1], mel10[,1])
29 |
30 | #matica pravidelnosti
31 | pravidelnosti21 <- cbind(PocTonov, mat.or.vec(po-od+1,k+1))
32 | for (i in od:po) {
33 |   for (stlpec in 1:(k+1)) {
34 |     pravidelnosti21[i-od+1,stlpec+1] <-
35 |       dim(as.matrix(TROJICE(i, melodie21[1:i, stlpec], c(1:i))))[1]
36 |     + dim(as.matrix(ROVNOBEZ(i, melodie21[1:i, stlpec], c(1:i))))[1]
37 |   }
38 | }
39 |
40 | #vykreslenie
41 | farby <- c("darkblue", "darkgoldenrod1", "forestgreen")
42 | legenda <- c('Pre□Elisku', 'Melodia□28', 'Melodia□10')
43 | plot(0,0,xlab="Pocet□tonov", ylab="Miera□pravidelnosti"
44 | ,xlim=c(od,po),ylim=c(0,28))
45 | for (i in 2:(k+2)) {
46 |   lines(PocTonov, pravidelnosti21[,i], col=farby[i-1], lwd=2)}
47 | title("Porovnanie□pravidelnosti")
48 | legend('topleft', legenda, lty=1, lwd=2, col=farby, bty='n',
49 | cex=1.25)
50 |
51 | #####porovnanie dalsich nahodnych melodii s Eliskou, od 10 po 35
52 | tonov#####
53 | od <- 10
54 | po <- 35
55 | n <- po
56 | PocTonov <- c(od:po)
57 |
58 | #generovanie matice melodii, 1.stlpec Eliska, 'k' dalsich nahodne
59 | k <- 5
60 | nahodne35 <- mat.or.vec(n,k)
61 | for (i in 1:k) nahodne35[,i] <- sample(1:n, n, replace=TRUE)
62 | melodie35 <- cbind(Eliska, nahodne35)
63 | write.table(data.frame(melodie35), "d:/melodie35.txt")
64 |
65 | #matica pravidelnosti
66 | pravidelnosti35 <- cbind(PocTonov, mat.or.vec(po-od+1,k+1))
67 | for (i in od:po) {
68 |   for (stlpec in 1:(k+1)) {
69 |     pravidelnosti35[i-od+1,stlpec+1] <-
70 |       dim(as.matrix(TROJICE(i, melodie35[1:i, stlpec], c(1:i))))[1] +
71 |       dim(as.matrix(ROVNOBEZ(i, melodie35[1:i, stlpec], c(1:i))))[1]
72 |   }
73 | }
74 |

```

```

75 #vykreslenie
76 farby <- c("darkblue", "gold", "forestgreen", "darkorange2",
77 "dodgerblue1", "red2")
78 legenda <- c('Pre_Elisku', 'Nahodna_1', 'Nahodna_2', 'Nahodna_3',
79 'Nahodna_4', 'Nahodna_5')
80 plot(0,0,xlab="Pocet_tonov", ylab="Miera
81 pravidelnosti",xlim=c(od,po),ylim=c(0,78))
82 for (i in 2:(k+2)) {
83 lines(PocTonov, pravidelnosti35[,i], col=farby[i-1], lwd=2)}
84 title('Porovnanie_pravidelnosti_pre_dlhsie_melodie')
85 legend('topleft', legenda, lty=1, lwd=2, col=farby, bty='n',
86 cex=1.25)
87
88 #####porovnanie s Nocturnom#####
89 NocturnoTony <- c('be4','c5','cis5','a4','be4','fis4','f4','f4',
90 'f4','f4','fis4','f4','dis4','c4','cis4','be3','be4','c5','cis5',
91 'a4','be4','a4','gis4','a4','c5','be4','fis4','f4','fis4','e4',
92 'f4','be4','a4','gis4','g4')
93
94 n <- length(NocturnoTony)
95 por <- c(1:n)
96 Nocturno <- rep(0,n)
97 for (i in 1:n) Nocturno[i] <- which(stupnica==NocturnoTony[i])
98 KRESLI(Nocturno)
99
100 od <- 10
101 po <- 35
102 n <- po
103 PocTonov <- c(od:po)
104
105 #generovanie matice melodii, 1.stlpec Eliska, 'k' dalsich nahodne
106 k <- 5
107 nahodne35 <- mat.or.vec(n,k)
108 for (i in 1:k) nahodne35[,i] <- sample(1:n, n, replace=TRUE)
109 melodie35 <- cbind(Nocturno, nahodne35)
110 write.table(data.frame(melodie35), "d:/melodie35druhe.txt")
111
112 #matica pravidelnosti
113 pravidelnosti35 <- cbind(PocTonov, mat.or.vec(po-od+1,k+1))
114 for (i in od:po) {
115   for (stlpec in 1:(k+1)) {
116     pravidelnosti35[i-od+1,stlpec+1] <-
117       dim(as.matrix(TROJICE(i, melodie35[1:i, stlpec], c(1:i))))[1] +
118       dim(as.matrix(ROVNOBEZ(i, melodie35[1:i, stlpec], c(1:i))))[1]
119   }
120 }
121
122 #vykreslenie
123 farby <- c("darkblue", "gold", "forestgreen", "darkorange2", "dodgerblue1",
124 "red2")
125 legenda <- c('Nocturno_Op._9_No.1', 'Nahodna_1', 'Nahodna_2',
126 'Nahodna_3', 'Nahodna_4', 'Nahodna_5')
127 plot(0,0,xlab="Pocet_tonov", ylab="Miera_pravidelnosti",
128 xlim=c(od,po),ylim=c(0,83))
129 for (i in 2:(k+2)) {
130 lines(PocTonov, pravidelnosti35[,i], col=farby[i-1], lwd=2)}
131 title('Porovnanie_pravidelnosti_pre_dlhsie_melodie')
132 legend('topleft', legenda, lty=1, lwd=2, col=farby, bty='n',
133 cex=1.0)

```

Spoločný kód *funkcie.r* - tieto funkcie používajú ostatné programy

```

1 #####
2
3 KRESLI <- function(vysky)

```

```

4 | {
5 |   por <- c(1:length(vysky))
6 |   n <- max(max(vysky),length(vysky))
7 |   plot(5,
8 |        5,
9 |        type="n",
10 |        ann = FALSE,
11 |        axes=FALSE,
12 |        xlim=c(1,n),
13 |        ylim = c(1,n))
14 |
15 |   abline(v=seq(1,n,1), h=seq(1,n,1), col = "black")
16 |
17 |   points(por,
18 |          vysky,
19 |          col="black",
20 |          pch=19)
21 | }
22 |
23 | #####
24 |
25 | #ak su tri body na priamke, obsah trojuholnika ktory tvoria
26 | #je 0
27 | #determinant je obsah
28 |
29 | TROJICE <- function(n, vysky, por)
30 | {
31 |   troj <- matrix( c(-1,-1,-1),nrow=1,ncol=3)
32 |   for (i in 1:(n-2))
33 |   {
34 |     for (j in (i+1):(n-1))
35 |     {
36 |       for (k in (j+1):n)
37 |       {
38 |
39 |         M <- t( matrix(data=c(i,j,k,
40 |                               vysky[i],vysky[j], vysky[k],
41 |                               1,1,1),nrow=3,ncol=3) )
42 |
43 |         if (round(det(M),3) == 0)
44 |         {
45 |           #maju rovnaku dlzku?
46 |           if (sqrt((por[k]-por[j])^2 + (vysky[k]-vysky[j])^2)
47 |              == sqrt((por[j]-por[i])^2 + (vysky[j]-vysky[i])^2 ))
48 |             troj <- rbind( troj , sort(c(por[i],por[j], por[k])) )
49 |         }
50 |       }
51 |     }
52 |   }
53 |   Tr <- unique(troj[-1,])
54 |   return(Tr)
55 | }
56 |
57 |
58 | #####
59 |
60 | ROVNOBEZ <- function(n, vysky, por)
61 | {
62 |   #vytvorenie matic rozdielov (Xroz a Yroz)
63 |
64 |   Yroz <- matrix(rep(-1000,length.out=(n-1)*(n-1)), nrow=n-1,
65 |                  ncol=n-1)
66 |
67 |   for (i in 1:(n-1))
68 |   {
69 |     for (j in (i+1):n)

```

```

70 | {
71 |   Yroz[j-1,i] <- vysky[j]-vysky[i]
72 | }
73 | }
74 |
75 | #dvojice riadkov v suradnice ktore maju:
76 | # 1.cislo rovnake ako aktual[1]
77 | # 2. cislo o 1 vacsie ako aktual[1]
78 | # 1.cislo o 1 mensie ako aktual[2]
79 | # 2. cislo rovnake ako aktual[1,]
80 | #neporovnavame lebo su to len 3 body
81 |
82 | # pri x-ovej vzdialenosti staci porovnavat ich poradove cisla
83 | # bod na [5,3] v Ynove je 6.-3. bod
84 |
85 | #do rovnobezniky sa ukladaju stvorice bodov (skut. por. cisel)
86 | #rovnobeznikov
87 |
88 | rovnobezniky <- matrix( c(-1,-1,-1,-1),nrow=1,ncol=4)
89 |
90 | for (i in (1-n):(n-1))
91 | {
92 |   suradnice <- which(Yroz==i, arr.ind=TRUE)
93 |   #nie abs(Yroz! lebo chcem zachovat smer)
94 |   if (dim(suradnice)[1]>0)
95 |   {
96 |     for (j in 1:dim(suradnice)[1])
97 |     {
98 |       aktual <- suradnice[j,]
99 |       #vyhodime tie s ktorym ma zhodnu niektoru suradnicu
100 |       suri <- suradnice[suradnice[,1] != aktual[1] & suradnice[,2]
101 |       != (aktual[1]+1) & suradnice[,1] != (aktual[2]-1)
102 |       & suradnice[,2] != aktual[2],]
103 |
104 |       if ((dim((as.matrix(suri)))[1]) !=0)
105 |       {
106 |
107 |         if((dim((as.matrix(suri)))[2]) !=1)
108 |         {
109 |
110 |           #otestuje ci su Xove rovnake
111 |           for (k in 1:dim(t(as.matrix(suri)))[1])
112 |           {
113 |             if ((aktual[1]+1 - aktual[2] ==
114 |             abs((as.matrix(suri))[k,1]+1 - (as.matrix(suri))[k,2])))
115 |             rovnobezniky <- rbind(rovnobezniky, sort(c(suri[k,1]+1,
116 |             suri[k,2], aktual[1]+1, aktual[2])))
117 |             #+1 tam nemusi byt lebo rozdiel to nezmeni
118 |           }
119 |         }
120 |
121 |         else
122 |         {
123 |           if (aktual[1]+1 - aktual[2] == abs(suri[1]+1 - suri[2]))
124 |           rovnobezniky <- rbind(rovnobezniky, sort(c(suri[1]+1
125 |           ,suri[2], aktual[1]+1, aktual[2])))
126 |           #+1 tam nemusi byt lebo rozdiel to nezmeni
127 |         }
128 |       }
129 |     }
130 |   }
131 | }
132 |
133 | }
134 |

```

```

135 #1. riadok su -1 tky, vyhodime duplikaty
136 return(unique(rovnobezniky[-1,]))
137 }
138
139
140 #####
141
142 COSTAS_GOLOMB <- function(n, vysky)
143 {
144   golomb <- rep(0,n)
145   kons <- 2*n-2
146   for (i in 1:n) golomb[i] <- (i-1)*kons+vysky[i]
147   #urcite G. pravitko ak vysky z Costasovho pola
148   dlzky <- c(-1)
149
150   for (i in 1:(n-1)) {
151     #napocitame dlzky na pravitku
152     for (j in (i+1):n) {
153       dlzky <- cbind(dlzky, c(golomb[j]-golomb[i]))
154     }
155   }
156   dlzky <- dlzky[-1]
157   dobre <- c(kons)
158
159   for (d in 1:(2*n-3)) {
160     for (i in 1:n) golomb[i] <- (i-1)*(kons-d)+vysky[i]
161     dlzky <- c(-1)
162     for (i in 1:(n-1)) {
163       for (j in (i+1):n) {
164         dlzky <- cbind(dlzky, c(golomb[j]-golomb[i]))
165       }
166     }
167     dlzky <- dlzky[-1]
168     if (length(dlzky) == length(unique(dlzky))) dobre <-
169     cbind(dobre, kons-d)
170   }
171
172   kons <- min(dobre)
173   for (i in 1:n) golomb[i] <- (i-1)*kons+vysky[i]
174   return(golomb)
175 }

```

A.3 Zdrojové kódy v jazyku MiniZinc

Program *golomb.mnz*

```

1  include "alldifferent.mzn";
2
3  int: m=20;      % pocet bodov na pravitku
4  int: nmax=290; % ak vyjde nepripustnost, treba zvyssit nmax
5
6  var int: n;    % posledny bod na pravitku, ma byt <= nmax
7  (aby boli premenne ohranicene)
8  array[1..m] of var 1..nmax: f; % body na pravitku
9
10 constraint n = max([f[i] | i in 1..m]);
11 constraint n <= nmax;
12
13 constraint f[1]=1;
14 constraint f[m]=n;
15

```



```
16 | constraint ((f[2]-f[1]) < (f[m]-f[m-1]));
17 | constraint forall(i in 2..m) (f[i-1] < f[i]);
18 | constraint alldifferent([f[j]-f[i] | i,j in 1..m where i<j]);
19 | % staci i<j namiesto not(i=j)
20 |
21 | % ULOHA 1: najdeme optimalne n
22 | solve minimize n;
23 |
24 | % ULOHA 2: s optimalnym n dame v konfiguracii
25 | 'print_all_solutions';
26 | %constraint n=25;
27 | %solve satisfy;
```

Príloha B

B.1 Výstupy programu hudba.r

Melódia Costas (počet rovnobežníkov: 0, počet ekvidistančných trojíc: 0):

Por. číslo	Pôvodný	Posunutý vektor	Golombovo	vektor d v dvoj-
tónu	vektor v	v v notách	pravítko	kovej sústave
1	5	cis3	5	100000
2	2	be2	37	101011
3	10	fis3	80	11101
4	4	c3	109	110011
5	20	e4	160	10111
6	8	e3	183	101100
7	17	cis4	227	100010
8	16	c4	261	11110
9	11	g3	291	100001
10	9	f3	324	110000
11	22	fis4	372	11111
12	18	d4	403	100110
13	21	f4	441	11011
14	13	a3	468	101001
15	19	dis4	509	10011
16	3	h2	528	101111
17	15	h3	575	11010
18	6	d3	601	100100
19	7	dis3	637	101000
20	12	gis3	677	100101
21	14	be3	714	10110
22	1	a2	736	0

Melódia 10 (počet rovnobežníkov: 9, počet ekvidistančných trojíc: 1):

Por. číslo	Pôvodný	Posunutý vektor	vektor d v dvoj-
tónu	vektor v	v v notách	kovej sústave
1	6	d3	100011
2	11	g3	101000
3	5	cis3	10110
4	2	be2	100011
5	11	g3	11101
6	11	g3	10111
7	19	dis4	100001
8	10	fis3	11011
9	19	dis4	101000
10	9	f3	100000
11	5	cis3	11101
12	19	dis4	101100
13	1	a2	11111
14	13	a3	11101
15	18	d4	11100
16	6	d3	11101
17	4	c3	101010
18	14	be3	11101
19	13	a3	101100
20	1	a2	100110
21	5	cis3	11100

Melódia 28 (počet rovnobežníkov: 22, počet ekvidistančných trojíc: 6):

Por. číslo	Pôvodný	Posunutý vektor	Vektor d v dvoj-
tónu	vektor v	v v notách	kovej sústave
1	16	c4	11111
2	14	be3	101111
3	5	cis3	11011
4	3	h2	11101
5	1	a2	101000
6	1	a2	101100
7	1	a2	110000
8	8	e3	11001
9	15	h3	100111
10	5	cis3	11010
11	5	cis3	100110
12	6	d3	101111
13	1	a2	100011
14	2	be2	101110
15	15	h3	11110
16	11	g3	101000
17	4	c3	10111
18	8	e3	10110
19	6	d3	11100
20	14	be3	100000
21	15	h3	101101

Príloha C

C.1 Formulár k experimentu

Vypočuj si tri melódie.

1. Aký si hudobník?

- a) Nehrám aktívne na žiadny hudobný nástroj
- b) Hrám na nejaký hudobný nástroj
- c) Som profesionálny hudobník alebo pedagóg

2. Zoraď melódie podľa toho ako sa Ti páčia

	Melódia 1	Melódia 2	Melódia 3
Najviac sa mi páči	☐	☐	☐
Menej sa mi páči	☐	☐	☐
Najmenej sa mi páči	☐	☐	☐

3. Na škále 0-10 vyjadri ako výrazný je rozdiel medzi melódiami

Každej melódii daj číslo od 0 po 10 - čím vyššie číslo, tým viac sa Ti melódia páči. Zároveň vyjadri o koľko sa viac sa Ti páči oproti ostatným. Ak sa Ti jedna páči oveľa viac ako druhá, daj jej podstatne vyššie hodnotenie. Ak sa Ti nejaké páčili približne rovnako, daj im takmer rovnaké hodnotenia.

	0	1	2	3	4	5	6	7	8	9	10
Melódia 1	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Melódia 2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Melódia 3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐