

**Vysoká škola báňská – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky**

**Analýza neznámých textů  
Analysis of Unknown Texts**

**2016**

**David Rapaň**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student: **David Rapaň**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Analýza neznámých textů**  
**Analysis of Unknown Texts**

Jazyk vypracování: čeština

### Zásady pro vypracování:

Problematika analýzy textů je v poslední době velmi důležitým tématem. Lze analyzovat sms zprávy, maily atp. Práce se bude zabývat analýzou neznámých textů se zaměřením na Voynichův rukopis. V rámci práce budou provedeny základní analýzy a vizualizace získaných výsledků struktury rukopisu s označením jak lze dále pokračovat pomocí nekonvenčních metod jako je fraktální geometrie atp.

### Struktura práce bude následující:

1. Zhodnocení současného stavu na poli analýzy textů.
2. Stanovení klasických ale i nekonvenčních oblastí analýzy.
3. Navrhnout vlastní řešení vybraného problému.
4. Vytvořte program a otestujte jej.
5. Zhodnoťte výsledky v závěru.

### Seznam doporučené odborné literatury:

- [1] Udo Kuckartz, Qualitative Text Analysis: A Guide to Methods, Practice and Using Software, SAGE Publications Ltd, 2014
- [2] Zelinka Ivan, Evoluční výpočetní techniky - principy a aplikace, BEN, Praha, 2008
- [3] Lenkova J., Voynichův rukopis aneb Nejzáhadnější kniha světa, CAD PRESS, 2012

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **prof. Ing. Ivan Zelinka, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry




prof. RNDr. Václav Šnašel, CSc.  
děkan fakulty

# Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *18. dubna 2016*

  
.....  
podpis studenta

# Poděkování

Velmi rád bych poděkoval prof. Ing. Ivanu Zelinkovi, Ph.D. za odbornou pomoc, vstřícný a motivující přístup, trpělivost a konzultace při vytváření této bakalářské práce.

# Abstrakt

Cílem práce je navrhnout a implementovat software, který by neměl být specializovaný na konkrétní neznámý text a bude schopen text přečíst a následně provést základní metody analýzy. V tomto konkrétním případě se bude jednat o přepis Voynichova rukopisu „Nejzáhadnější kniha světa“, která nebyla doposud rozšifrována ani největšími specialisty.

Bakalářská práce v teoretické části popisuje metody analýz nejen neznámých textů, ale analýzy všeobecně. Stanoví několik klasických metod a k závěru ukáže možný postup nekonvenční metodou fraktální geometrie.

Praktická část je zaměřena na návrh software a jeho implementaci na platformě .NET v jazyce C#. Software umožní přepis přečíst, určit jeho nejasnosti a následně je opravit vůči celkovému textu. Poté bude schopen zobrazit jednoduché analýzy nad strukturou textu a zobrazit je.

## Klíčová slova

analýza, neznámý text, Voynich, fraktální geometrie

# Abstract

The goal of this thesis is design and implementation of a software which wouldn't be specialized on a specific unknown text. Will be able to read and perform basic analysis methods. In this particular case it's going to be transcript of Voynich manuscript "The World's Most Mysterious Manuscript". This manuscript wouldn't have decrypted yet.

Bachelor thesis in theoretical part describe methods of analysis texts. Define several classic methods and in the end will show possible process unconventional method of fractal geometry.

Practical part of this thesis is based on design software and his implementation on the .NET platform in C#. Software will allow to read the transcript, determine his unclear parts and repair this parts which are compared to whole text.

## Key words

analysis, unknown text, Voynich, fractal geometry

## Seznam použitých symbolů a zkratek

Zkratka	Význam
---------	--------

---

FSG	First Study Group
-----	-------------------

---

# Obsah

1 Úvod .....	1
1.1 Cíl práce .....	1
1.2 Obsah kapitol.....	2
2 Teoretická část .....	3
2.1 Analýza .....	3
2.1.1 Kvalitativní a kvantitativní analýza .....	3
2.1.2 Výzkum v kvalitativním směru .....	3
2.1.3 Problémy kvalitativní analýzy dat v praxi.....	4
2.1.4 Motivace .....	4
2.2 Současný stav .....	4
2.2.1 Klasická hermeneutika .....	5
2.2.2 Zakotvená teorie .....	6
2.2.3 Kategorie .....	6
2.2.4 Kód .....	7
2.3 Základní metody analýzy textu.....	7
2.3.1 Základní podobnosti a rozdíly tří metod .....	8
2.3.2 Tematická kvalitativní analýza textu .....	8
2.3.3 Evaluační kvalitativní analýza textu .....	9
2.3.4 „Type-Building“ analýza textu.....	10
2.4 Fraktální geometrie v kryptologii .....	11
2.4.1 Kryptologie .....	11
2.4.2 Nekonvenční metoda .....	12
2.5 Nejzáhadnější rukopisy světa .....	12
2.5.1 The Book of Soyga .....	13
2.5.2 Codex Seraphinianus.....	13
2.5.3 Hypnerotomachia Poliphili.....	14
2.5.4 The Oera Linda Book .....	15
2.5.5 The Ripley Scrolls .....	15
2.5.6 The Smithfield Decretals .....	16
2.5.7 The Rohonc Codex .....	16
2.5.8 The Red Book .....	17
2.5.9 Prodigiorum ac ostentorum chronicon .....	18
2.6 Voynichův rukopis .....	19
2.6.1 Obsah .....	20
2.6.2 Rukopis.....	21
2.6.3 Přepis.....	21
3 Praktická část.....	22
3.1 Architektura řešení.....	22

3.2	Knihovna Transcript.....	22
3.3.1	Třída Constants .....	23
3.3.2	Třída Cell .....	23
3.3.3	Třída Word .....	25
3.3.4	Třída Page.....	26
3.3.5	Třída Book .....	26
3.4	Knihovna Statistics .....	29
3.4.1	Třída AnalyzedWord.....	29
3.4.2	Třída Analysis .....	29
3.5	GUI (WinForms) – AnalysisUnknownTexts.....	30
3.5.1	Formulář MainForm .....	30
3.5.2	Formulář DetailCellForm .....	30
3.5.3	Formulář DetailAnalyzedWordForm .....	31
3.5.4	Formulář BookMapForm.....	31
3.5.5	Formulář HistogramForm.....	31
3.5.6	Formulář DataGridViewForm .....	31
3.6	Použití aplikace (programu) .....	31
3.6.1	Hlavní formulář (MainForm) .....	31
3.6.2	Formulář detailu analyzovaných dat (DetailAnalyzedWordForm).....	32
3.7	Ukázka aplikace (programu).....	32
3.7.1	Hlavní formulář (MainForm) .....	32
3.7.2	Formulář detailu buněk (DetailCellForm) .....	33
3.7.3	Formulář detailu analyzovaných dat (DetailAnalyzedWordForm).....	34
3.8	Výsledky.....	34
3.8.1	Histogram.....	35
3.8.2	„Mapa“ .....	35
4	Závěr.....	38
	Použitá literatura.....	40
	Seznam příloh.....	- 42 -

## Seznam obrázků

Obrázek 1: Přístup hermeneutiky [1] .....	6
Obrázek 2: Proces kvalitativní analýzy textu v rámci kategorií [1] .....	7
Obrázek 3: Proces tematické analýzy textu [1].....	9
Obrázek 4: Proces evaluační analýzy textu [1] .....	10
Obrázek 5: Proces „Type-Building“ analýzy textu [1].....	11
Obrázek 6: „Book of Soyga“ – ukázka jedné z 36 posledních stránek [9].....	13
Obrázek 7: „Codex Seraphinianus“ ukázka z prapodivné knihy [10].....	14
Obrázek 8: „Hypnerotomachia Poliphili“ ukázka ilustrace s textem [11].....	15
Obrázek 9: „The Smithfield Decretals“ ukázka textu a ilustrace .....	16
Obrázek 10: „The Rohonc Codex“ ukázka textu s ilustrací .....	17
Obrázek 11: „The Red Book“ ukázka ilustrace knihy [16].....	17
Obrázek 12: „Prodigiorum ac ostentorum chronicon“ invaze kobylek [17] .....	18
Obrázek 13: „Prodigiorum ac ostentorum chronicon“ kosmická loď .....	18
Obrázek 14: Ukázka písma z Voynichova rukopisu .....	19
Obrázek 15: „Voynichův rukopis“ ilustrace z astronomické části rukopisu .....	20
Obrázek 16: Ukázka výskytu možných nejasností v přepisu .....	24
Obrázek 17: Možnosti jak může slovo vypadat s porovnáním vůči textu.....	27
Obrázek 18: Ukázka aplikace – hlavní formulář.....	32
Obrázek 19: Ukázka aplikace – detail buňky před vyjasněním textu - vlevo a po - vpravo .....	33
Obrázek 20: Ukázka aplikace – analyzované slovo - vlevo a znak - vpravo .....	34
Obrázek 21: Výsledky – histogram výskytu slov s počtem > 50 .....	35
Obrázek 22: Výsledky – hlavní výstup aplikace - Mapa knihy (zvětšení).....	36
Obrázek 23: Výsledky – hlavní výstup aplikace - Mapa knihy (bez zvětšení) .....	37
Obrázek 24: Statistická charakteristika výskytu znaků Voyničiny .....	38



# 1 Úvod

Analýza textů není jen o textu samotném, ale je to vlastně jistá forma sběru informací i o autorovi, o jeho pohledu na skutečnost, pojmu o světě a jeho osobu samotnou. Dovedeme tyto informace dokonce přirovnat například k otiskům prstů, kdy autor do textu otiskne kus sebe sama. Analýza nad takovým textem nám pak dokáže mnohé prozradit. A v době kdy prakticky vše převzala digitální forma a je jakákoliv publikace snadnější než kdy dříve nabírá tato vědní disciplína na důležitosti. Například se velmi rozmáhá diskuze nad sledováním lidí různými agenturami, kdy tedy údajně analyzují nejen chování, ale také různé texty od sms zpráv až po emaily a tzv. vyhodnocují „hrozby“. Fakt, že toto je velmi dobrý způsob jak získat cenné informace o komkoliv je nepopiratelný.

Lingvistika a použití matematiky v lingvistice se však vztahuje výhradně na přirozené jazyky, nikoliv šifry. K šifrá a kryptografii je třeba přistupovat i jinými metodami, jelikož většina z nich již počítá s porozuměním textu. Zde se naskytují metody, které porovnávají četnosti různých důležitých slov v textu. Kdy se například spojky vyskytují stejně pravidelně a rovnoměrně a slova k tématu zřídka a ne rovnoměrně a hlavně v rámci kapitol. Tyto a další vzorce se porovnávají s přirozenými jazyky, protože pod šifrou by takový jazyk měl být a jeho reprezentace není skrytá.

## 1.1 Cíl práce

Cílem práce je návrh a implementace nástroje schopného přečíst přepis neznámého textu (šifry), takto získaná data zpracovat vhodným způsobem a doplnit mezery mezi rukopisem a přepisem způsobené chybou čtení nebo nekvalitou rukopisu. Následně provést jednoduché druhy analýzy jako výčet použitých znaků neboli písmen a slov, jejich počet a následně zobrazit histogramy a mapu průchodu celým textem s vyobrazením vybraných slov. Tento výstup by měl být dále určen k použití s fraktální geometrií k nekonvenčním analýzám nad vybraným textem.

Praktická implementace spočívá ve vytvoření programu určeného ke čtení přepisu Voynichova rukopisu z textového souboru. Součástí FSG přepisu Williama Friedmana je jasný postup jak přepis číst. Tyto postupy je nutné dodržet, avšak se snahou program situovat pro znouvopoužitelnost i s jinými šiframi. Dále předvedeme aplikaci jednoduchých analýz nad takto získanými daty a jejich zobrazení či vizualizaci uživateli. Bereme v potaz, aby nás software průběžně informoval a zobrazoval výsledky prováděných operací. Obsahem programu nebude implementace algoritmů fraktální geometrie, tento postup vyžaduje další studii a v této práci zmíním jen možné způsoby jak fraktální geometrii aplikovat.

Analýza jako taková je velmi široký pojem a v této práci se budu zabývat porozuměním textu, který neodpovídá žádnému přirozenému jazyku a je považován za šifru. A poté šifrovaný text analyzovat v rámci použitých znaků, tvarů a podobností použitých slov a jejich výskyt v textu. Tato práce je v jisté formě příprava na hlubší studii použití fraktální geometrie jako nekonvenční metoda analýzy dat v této podobě.

## 1.2 Obsah kapitol

Ve druhé kapitole se zabývám seznámením čtenáře se současným stavem na poli analýzy textů a dat celkově, základním rozdělením na kvalitativní a kvantitativní analýzy. Dále stanovením několika klasických metod analýzy textů a dat a jejich příkladů použití a důležitosti této disciplíny v širokém spektru oborů. Důležité je také zmínit základní aspekty fraktální geometrie a její použití nejen pro naše účely. Závěrem pokládám za nezbytné věnovat pár odstavců problematice Voynichova rukopisu.

Třetí kapitola obsahuje praktickou část, kde se zabývám návrhem software a jeho implementaci pro platformu .NET 4.5, konkrétně s využitím knihovny Windows Forms. Nejprve shrneme problematiku a ujasníme účel vytvářeného software a posléze popíšeme konkrétní řešení s důrazem na použité algoritmy čtení, korektury textu a následné analýzy textu.

## 2 Teoretická část

Existuje mnoho pojednání o analýzách nejen textů, ale všeobecně a tato kapitola se snaží vytvořit přehled a uvést do problematiky analýz v širším spektru. Kapitola se dále přímo specializuje na analýzu textů a jejich metody, úvod do fraktální geometrie a její použití a stručné shrnutí faktů o Voynichově rukopisu.

### 2.1 Analýza

Pod pojmem analýza si lze představit celou řadu oblastí, kam různé druhy analýz patří. Ve své podstatě analýza znamená shrnutí a ujasnění všech známých faktů a poznatků o analyzovaném materiálu a všech dat, která se dají specifickými způsoby zjistit. Metody, jak takových výsledků dosáhnout jsou popsány dále v kapitole.

#### 2.1.1 Kvalitativní a kvantitativní analýza

Nežli se zaměříme přímo na analýzu neznámých textů, je třeba stručně deklarovat základní rozdělení. Zatímco kvantitativní analýza je úzce spjata s čísly, počty a statistikou a nalézá uplatnění převážně v ekonomice, tak kvalitativní analýzu není tak jednoduché definovat a má často odlišný význam v různých vědních disciplínách. Kvalitativní analýzy mají mnohem vyšší kontrast a řekněme, že budeme vycházet z definice, že v rámci kvalitativních analýz, se bude jednat o texty a obrazy. [1]

Striktně kvalitativních dat se týká přímo interpretace textu a hermeneutiky, kvantitativní zase statistických a matematických dat. Jejich spojením poté v prvním případě jednáme s počty slov a analýzy obsahu a ve druhém případě ohledně hledání a prezentaci smyslu v kvantitativních datech. [1]

#### 2.1.2 Výzkum v kvalitativním směru

Definice kvalitativního výzkumu jde daleko za prostou analýzu textu, přičemž existuje mnoho správných definic. Kvalitativní výzkum zahrnuje různé, občas velmi nekonvenční, metody a techniky. Existuje mnoho knih na téma kvalitativního výzkumu a většina z nich bude obsahovat úplně odlišný přístup. Snaha autorů o vytvoření nových možností, jak výsledků dosáhnout je zřejmá a správná, protože na tento druh výzkumu lze nahlížet z mnoha perspektiv, ale všechny mají něco společné [1].

Charakteristiky kvalitativního výzkumu [1]:

- 1) Použití více metod místo jedné
- 2) Vhodnost metod
- 3) Orientace na každodenní události a znalosti
- 4) Kontext jako průvodce principy
- 5) Reflektivní schopnost
- 6) Porozumění a objevování principů
- 7) Principy otevřenosti
- 8) Příklad analýzy jako počáteční bod
- 9) Konstrukce reality jako základ

- 10) Kvalitativní výzkum se týká textů
- 11) Objev a teorie jako cíl

V mnoha knihách, zabývajících se výzkumem, také naleznete velmi často porovnání mezi kvalitativním a kvantitativním přístupem. Kvalitativní charakter se tak vyskytuje ve výzkumu kvantitativním a naopak. Kvalitativní výzkum zahrnuje mimo jiné i část věnovanou kvantitativním datům, což obsahuje slova jako „často“, „zřídka“, „obvykle“, „typicky“, apod. Tím se dostáváme k závěru, že kvalitativní a kvantitativní metody se navzájem nevylučují a vznikají tak smíšené metody. [1]

### 2.1.3 Problémy kvalitativní analýzy dat v praxi

Nalézt opravdu funkční a konkrétní metody pro analýzu kvalitativních dat je docela obtížné a snahou je se vyhnout potížím spojených s kvalitativním výzkumem. Proces jak analýzu provést a jasné kroky jsou zřídka kdy podrobně popsány, tak, aby mohly být znovu použity za podobným účelem. [1]

Dopracování se skrze kvalitativní metody k nějakému závěru občas mylně nabádá k pocitu, že si výzkumníci mohou dělat, co chtějí, vytvářet velkolepé výklady jak je postup správný. A to vše bez jakéhokoli rizika vyvrácení jejich „správných“ závěrů. Tímto se otázka ohledně univerzálních a objektivních standardů kvality stává zastaralou. [1]

### 2.1.4 Motivace

Na začátku snad každého výzkumu je třeba znát odpověď na otázku „Proč?“. A vlastně i v našem případě je třeba si takovou otázku na začátku položit. Čeho vlastně v rámci práce chceme dosáhnout? A jak toho chceme dosáhnout? A zda bude mít celkově nějaký smysl. Jak chceme takovou otázku nalézt? Jaké metody použijeme?

Existují tři základní typy výzkumu, které s tímto souvisí: *základní*, *aplikovaný* a *evaluační výzkum*. Základní výzkum se používá při experimentálních metodách, testování, a určování hypotéz, obecně všechny tři typy metod mohou pracovat jak s kvalitativními tak kvantitativními metodami. Jinde je zase možné se dozvědět o čtyřech dalších studiích: *průzkum*, *popis*, *testování hypotéz* a *evaluační*. A znovu v obou případech analýz, kdy existuje možnost jak metody navzájem kombinovat. Kvalitativní metody je možno častěji nalézt v prvních dvou studiích a zpřístupňují obecný pohled na věc. Startovní bod pro všechny druhy studií je „otázka“, bez ní je těžké si výzkum představit. [1]

Výzkum se týká analýzy textu, kterému se nedá logicky porozumět, jelikož je šifrován. Otázka tedy spočívá v dopátrání se metod a způsobů jak na text tohoto typu nahlížet, které zatím nebyly použity a nebát se poučit se i z dalších odvětví vědy, jako je například fraktální geometrie, která už byla schopná zodpovědět již celou řadu otázek, na které odpovědi doposud nebyly.

## 2.2 Současný stav

Stanovit současný stav vývoje, výzkumu či jakéhokoli bádání a popsat aktuálně používané metody není nikdy jednoduché. Vytvořit takový souhrn a nejlépe ve stručné podobě z toho množství

knih a informací bývá ale často nezbytná součást jakéhokoliv výzkumu. Je třeba vědět odkud začít a na čem stavět novou metodologii.

Mnoho metod a technik analýzy textů bylo vyvinuto v různých oblastech vědy od společenských věd přes psychologii až po vzdělávání. Jejich vývoj měl určitá úskalí a některé jsou velmi známé jako například zakotvená teorie. [1]

### **2.2.1 Klasická hermeneutika**

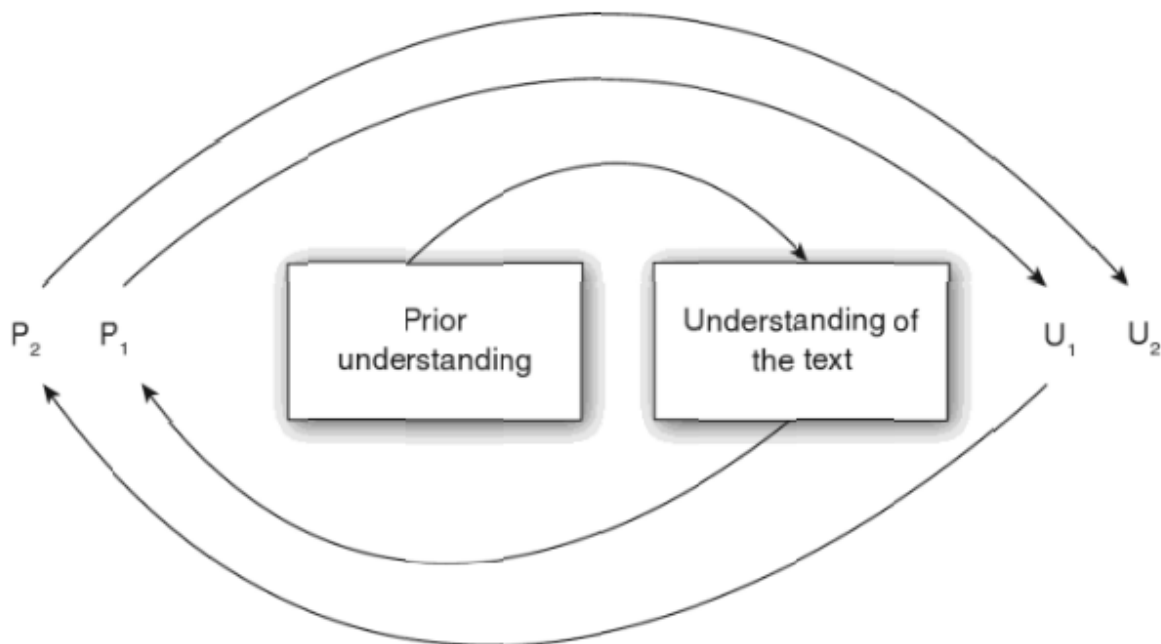
Hermeneutika se zabývá hledáním metod výkladu a porozumění světu, výkladem, vysvětlením a pochopením textu, událostí a faktů. Klasická hermeneutika částečně vychází z teologie a výkladu Písma svatého. I tak hermeneutika zapadá spíše do filosofie, ale má svůj přínos i pro analýzu textů.

Před samotnou analýzou dat je třeba se podívat na problém z širšího spektra a ne jen jako na text, který je třeba analyzovat a nějakým způsobem „vyřešit“. Pochopit smysl textu a jeho interpretaci a s tím pomůže hermeneutika.

Hlavní hermeneutický přístup [1]:

- 1) Podmínky při jakých byl text vytvořen
  - Kdo text psal, proč jej psal a komu jej psal.
- 2) Hermeneutický kruh (Obr. 1)
  - Text je interpretován jako suma jeho částí a jednotlivé části mohou být vysvětleny jen při porozumění celému textu.
- 3) Hermeneutický rozdíl
  - Znamená rozdíl mezi dorozumívajícími se prostředky, co známe a běžně používáme, a ty cizí, např. mezi rodnou řečí a řečí cizí země.
- 4) Přesnost a vhodnost
  - Přesnost porozumění typicky kulturním hodnotám a jejich pochopení.

Postup aplikovaný pro kvalitativní analýzu tedy záleží na motivaci týkající se výzkumu, pracovat s textem jako navzájem propojeným celkem, stanovení zda je text rozdílný oproti známým jazykům, při čtení být pozorný (neplatí samozřejmě u šifer) a stanovit rozdíly mezi poznáním a logikou aplikací kódovacích obrazů celku.



Obrázek 1: Přístup hermeneutiky [1]

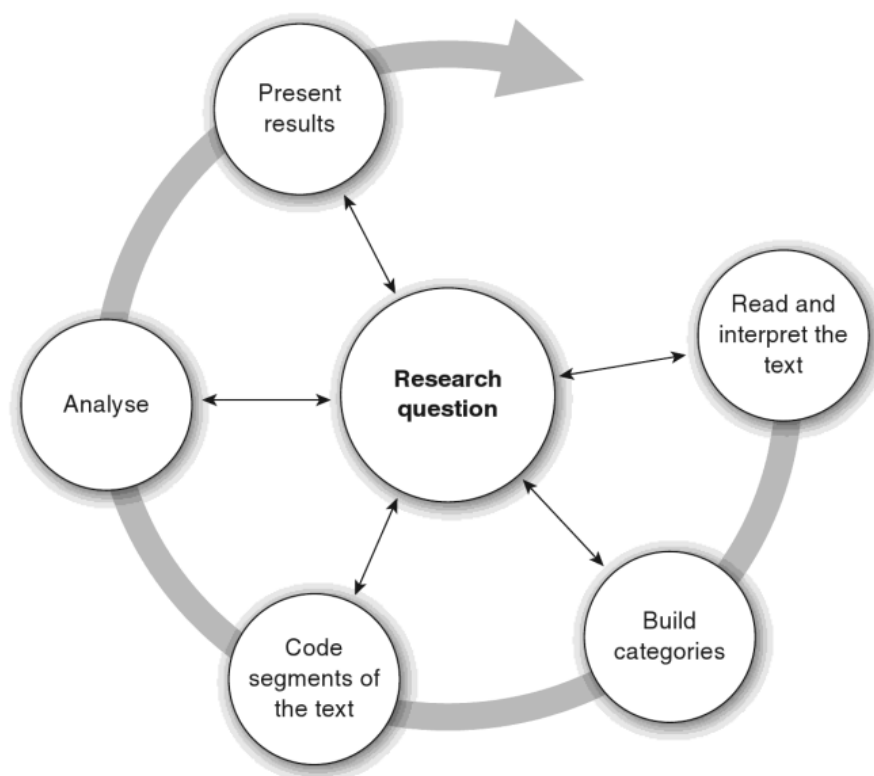
### 2.2.2 Zakotvená teorie

Je znovu metoda týkající se porozumění textu, což v jisté míře odporuje prvotnímu záměru řešit šifrovaný a neznámý text, kterému nejde rozumět, tudíž není až tak důležité se jí zaobírat do hloubky. Avšak cílem metody je vytvořit teorii. Cílem je probrat se daty a nechat vyplynout důležité aspekty řešených dat a postupně zjistit, co data znamenají v jejich širším smyslu. V této metodě hrají klíčovou roli *kategorie* a *kódování*, tyto termíny dále vysvětlím.

### 2.2.3 Kategorie

Termín kategorie má i v tomto kontextu význam, tak jak nám selský rozum napovídá. Jde o druh třídění do jednotlivých celků. Takto vytvořené entity mohou být lidé, instituce, nápady, objekty atp. V podstatě to znamená, že větší celek rozdělíme na menší části na základě jejich podobnosti. [1]

Tvoření kategorií je základní část jakékoliv aktivity týkající se analýz textu. Podobný systém klasifikace je možné pozorovat v každodenním životě jako vnímání světa kolem nás a to jak si jej vysvětlovat a celkově chápat. Podobný důvod má využití kategorií v kvalitativní analýze textu. Můžeme rozlišovat alespoň mezi pěti druhy kategorií (Obr. 2)



Obrázek 2: Proces kvalitativní analýzy textu v rámci kategorií [1]

## 2.2.4 Kód

Termín kód má 3 rozdílné popisy [1]:

- a) Jako šifrovací zařízení, jména a pozice jsou kódována ke skrytí informace.
- b) Kód jako nástroj o označování a indexování textu.
- c) Kód jako hodnota k určení množství částečných charakteristik.

Kódy jsou použity zvláště v zakotvené teorii, kde se vyskytuje v několika různých formách: otevřený kód, axiální kód a selektivní kód atp. Vyskytuje se v kvalitativním výzkumu jako analyzování, pojmenování, kategorizace, a teoretická organizace zpracovávaných dat, což odpovídá převážně bodu b).

## 2.3 Základní metody analýzy textu

Ohledně výzkumu zde máme různorodé metody, techniky a postupy, které nám pomohou dosáhnout úspěchu v rámci kvalitativních analýz. Pokusím se zde vysvětlit tři základní metody a to tematickou metodu, ohodnocující metodu a „type-building“ metodu.

### 2.3.1 Základní podobnosti a rozdíly tří metod

Všechny tři metody využívají tzv. kategorií, neboť základní analýza obsahu staví na tomto rozdělení při řešení problému a analyzování empirických dat příslušných kategorií. Metody na sobě také staví a vyplývají jedna z druhé a jsou si rovnocenné a vedou vždy ke konkrétnímu případu užití dle výzkumné otázky. Například není vždy výhodné použít „type-building“ metodu, jak se může na první pohled zdát. Velmi často je popisována jako jasný postup kvalitativního výzkumu. [1]

Podobnosti metod [1]:

- 1) Typ metod, které nemají prvotní požadavky na podobu a mohou být použity na stejná data
- 2) Komprimují data a vytváří jejich shrnutí
- 3) Jsou založeny na kategoriích
- 4) Mají jasně dané postupy použití a přesné definice
- 5) Aplikují se většinou na verbální data, ale mohou být použity i na obrázky či videa
- 6) Formulce zásad kvality pro všechny tři metody, rozlišení mezi dobrou a špatnou kvalitou analýzy

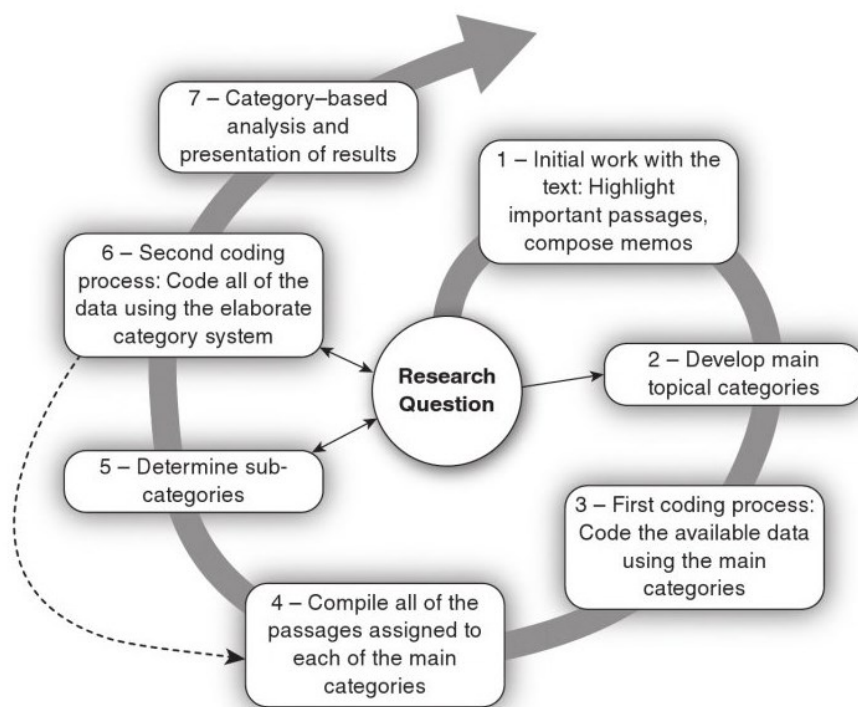
U následujících metod je možná aplikace dokonce před sesbíráním všech potřebných dat.

### 2.3.2 Tematická kvalitativní analýza textu

Široké spektrum metod může určit jak sestavit kategorie pro použití s tematickou metodou analýzy dat. Používají 2 postupy indukční a dedukční. Většina z nich prolíná oba, jen zřídka nalezneme metodu používající jen jeden druh postupu. Používá se víceúrovňový postup kategorizace: Na začátku se data kódují velmi nahrubo současně s hlavními kategoriemi (počet kategorií je na začátku velmi malý). V další fázi jsou kategorie dále děleny na základě dalších dat. Poté jsou data kompletně kódována znovu, analyzována podle nových kategorií a připravována na celkový souhrn. [1]

Tento typ metody se nejčastěji hodí na data typu „vyprávějíciho“ rozhovoru apod. (Obr. 3).



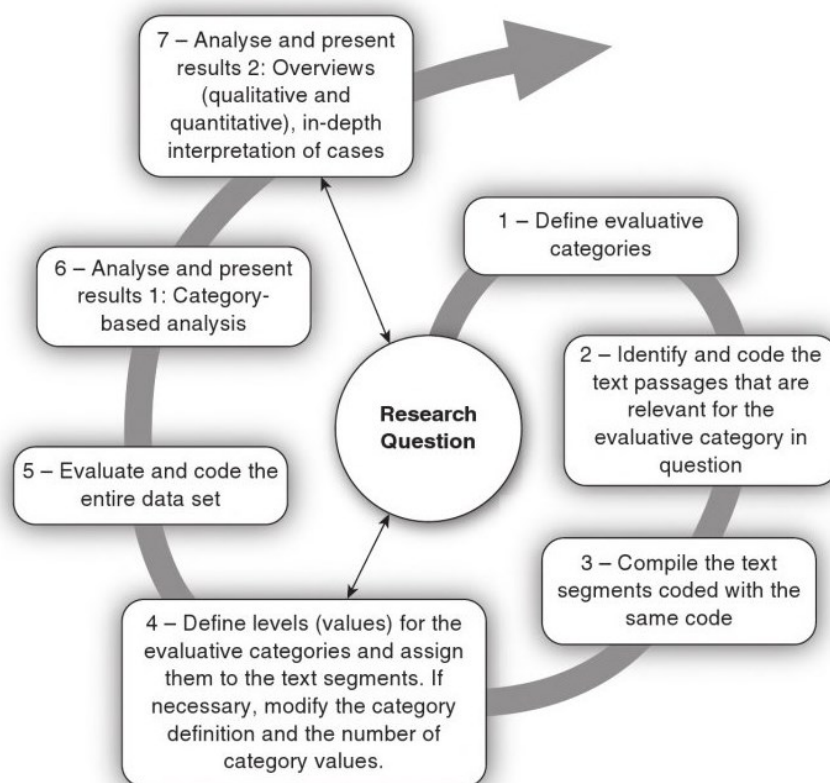


Obrázek 3: Proces tematické analýzy textu [1]

### 2.3.3 Evaluační kvalitativní analýza textu

Evaluační metoda je další ze základních metod systematické kvantitativní analýzy dat. Je použita v různých projektech týkajících se empirického výzkumu. Oproti tematické analýze, která se soustředí na identifikaci, systematizaci a analýzu témat a subtémat a jak jsou propojeny. Evaluační metoda zahrnuje posuzování, třídění a hodnocení obsahu. [1]

Posuzují se data a staví kategorie s charakteristikou k číslování a stupňování. Například, respondent je posuzován použitím evaluační kategorie „sebevědomí“, což obsahuje podkategorie „vysoké sebevědomí“, „průměrné sebevědomí“ a „nízké sebevědomí“. Postup je znázorněn na Obr. 4. [1]

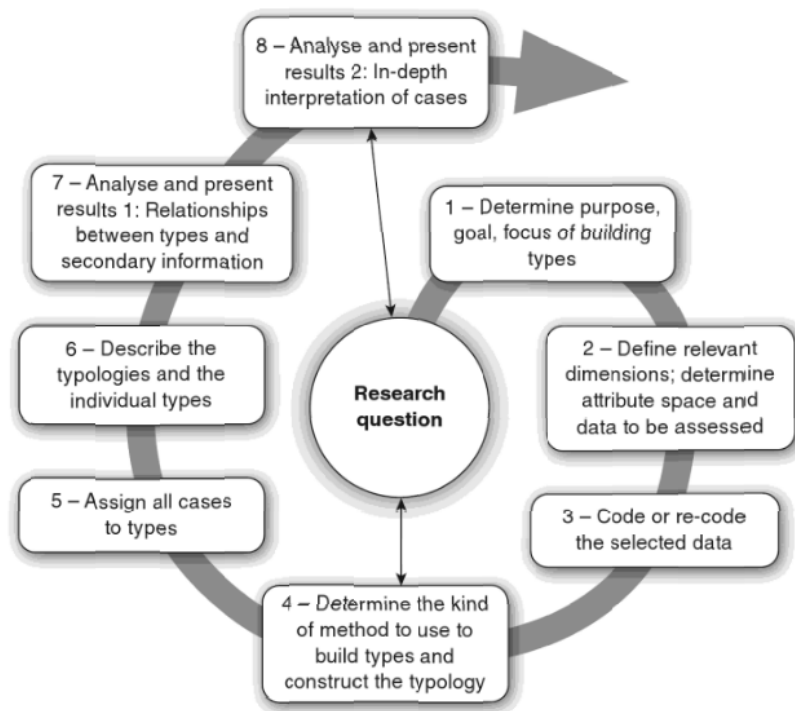


Obrázek 4: Proces evaluační analýzy textu [1]

### 2.3.4 „Type-Building“ analýza textu

Podle mnoha metodologů, vytváření typů a vývoj typologie je hlavní cíl kvalitativní analýzy dat (textu). Tento typ metody je velmi komplexní a vychází z předchozích dvou metod, respektive zakládá na nich. Účel metody zahrnuje vyhledání multidimenzionálních vzorců a modelů, což umožní pochopení složitých dat. [1]

Elementy jsou ve skupinách podle typů odpovídajících jejich podobnostem vybraných atributů. Takže v rámci empirického výzkumu „type-building“ odpovídá vytváření skupin podle určitých vzorců nebo skupin, které se liší od ostatních, co je obklopují. Všechny typy, které jsou použité k popisu jevu, nazýváme typologií. Typologie vždy obsahuje různé typy a jejich vztahy mezi sebou, jejich struktury podle podobností a rozdílů. Použití na Obr. 5. [1]



Obrázek 5: Proces „Type-Building“ analýzy textu [1]

Proces této metody se se liší oproti předchozím v mnoha aspektech. Je možné sestavit několik různých topologií v rámci jednoho projektu. Například, jedna může ukázat jak je informace zpracovávána oproti atomové energii a jiná zase demonstrovat jak byly zhodnoceny rizika užití atomové energie po katastrofě ve Fukushima.

## 2.4 Fraktální geometrie v kryptologii

Nebudu se zde zabývat historií ani popisem fraktální geometrie o tom pojednává mnoho knih a učebnic, spíše se zaměřím na nástin jejího využití, ačkoliv je fraktální geometrie stále velmi mladá věda. Její schopnost „předpovědi“ a predikcí je již delší dobu známá, použití v generování prostředí atp. Fraktály můžeme nalézt snad ve všech vědních disciplínách, kdy se využívají ve fyzice, chemii, biologii, ekonomice a také medicíně. Poslední výskyt se objevuje v herním prostředí. Herní vývojáři se začínají specializovat na průběžné generování prostředí právě pomocí fraktální geometrie. Již dnes existuje typ hry, kdy se celé prostředí kolem hráče generuje právě tehdy, kdy je potřeba. Plyne z toho úspora na velikosti a také různorodost a rozmanitost.

Problematika je opravdu komplexní a věřím, že fraktální geometrii čeká v budoucnu ještě velmi široké uplatnění. Další způsoby se snažím dále nastínit.

### 2.4.1 Kryptologie

Věda zabývající se šifrováním se rozděluje na kryptografii, která řeší šifrování a metody šifrování a kryptoanalýzu, která je opakem a zabývá se dešifrováním a metodami luštění šifer.

Nejznámější šifrovací stroj je Enigma, hojně používaný nejen za 2. světové války. A oproti Enigmě Turingův Bombe, který jí byl schopný do určité míry dešifrovat.

V knize *Fraktální geometrie – principy a aplikace* (2006) jsou popsány různé postupy a jak k šifrování pomocí fraktální geometrie a neuronových sítí přistupovat. Řešení jak metodou fraktální geometrie přistupovat ke kryptoanalýze však nikoli. Při hledání jsem nenašel žádné relevantní informace týkající se použití fraktálů při řešení šifer, ale pomáhají při tolika jiných různých odvětvích, tak proč by nemohli i tady.

## 2.4.2 Nekonvenční metoda

Použití fraktálů k analýze textu a k tomu zašifrovaného se jeví jako nelehký úkol, zvláště když se jedná o tak mladé odvětví. Zde se nabízí několik druhů přístupu. Výstup implementovaného programu v této práci je něco na způsob mapy průchodu celou knihou s vytyčením význačných slov a jako celek tvoří nehomogenní mřížku. Tato mřížka by šla dále analyzovat, avšak musela by se upravit do konstantního počtu buněk v řádku a poté pokračovat například použitím programu HarFA [8]. Který je schopný homogenní mřížky o různých velikostech (viz. Web) analyzovat.

Jako další možnost se jeví implementace vlastního nástroje, který staví na principech výše zmíněného programu HarFA [8]. S použitím analýzy struktur fraktální geometrií z pohledu přímo na rukopis neznámého textu.

Použití neuronovou sítí na naučení rozpoznávání Voynichova rukopisu se jeví jako další způsob bádání a dále v rukopise hledat určité vzorce, třeba například dokonce sobě-podobnost či sobě-příbuznost a dokázat tak třeba vyvrátit chybné teorie o smysluplnosti rukopisu nebo naopak. Stejně, jak široké spektrum aplikací fraktální geometrie zaobírá, tak by zde bylo možné vyzkoušet stejné množství metod, jak provést analýzu tohoto typu. Ačkoliv je možné, že interpretovaná data nebudou až tolik čitelná, jak by se mohlo zdát, jelikož je fraktální geometrie stále velmi mladá věda.

## 2.5 Nejzáhadnější rukopisy světa

Na světě existuje asi 10 knih, u kterých vlastně nikdo neví, co se v nich nachází a o čem, že jsou. Pomyslnou knihou číslo jedna je v seznamu Top 10 Voynichův rukopis, avšak za zmínku stojí všechny:

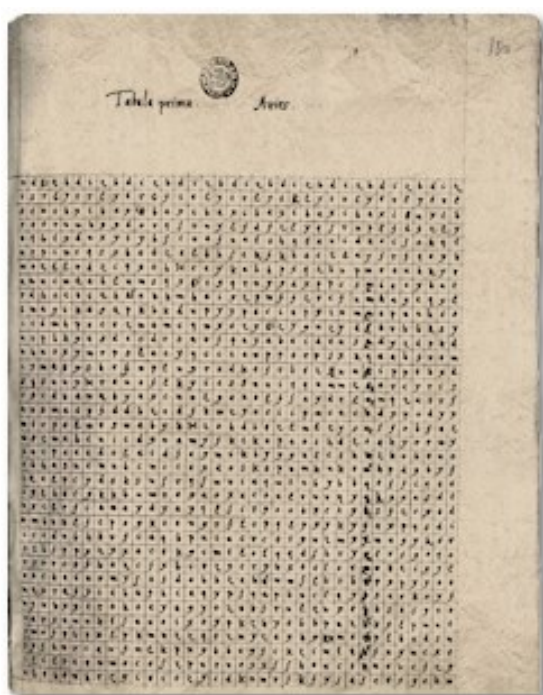
- 1) The Voynich Manuscript (Autor neznámý)
- 2) Prodigiorum ac ostentorum chronicon (Conrad Lycosthenes)
- 3) The Red Book (Carl Gustav Jung)
- 4) The Rohonc Codex (Autor neznámý)
- 5) The Smithfield Decretals (Autor neznámý)
- 6) The Ripley Scrolls (George Ripley)
- 7) The Oera Linda Book (Autor neznámý)
- 8) Hypnerotomachia Poliphili (Francesco Colonna)
- 9) Codex Seraphinianus (Luigi Serafini)
- 10) The Book of Soyga (Autor neznámý)

Dále se věnuji stručnému popisu těchto záhadných knih.

### 2.5.1 The Book of Soyga

Také známá pod názvem „Aldaraia“ zahrnuje jakési pojednání v latinském jazyce o magii z 16 století. Jednu dobu měl knihu v držení John Dee, který byl jejím obsahem uchvácen. Představa skutečné magie a okultismu ho fascinovala. Avšak nebyl za svého života schopen knihu rozluštit. Po jeho smrti kniha zmizela a objevila se až v roce 1692. [9]

A právě nejzajímavějších je posledních 36 stránek, které kniha obsahuje. Každá zobrazuje mřížku o 36 řádcích a 36 sloupcích (Obr. 6). Mřížka je vyplněna latinskými písmeny k celkovému počtu 46 656 znaků. Odpovídající text je jen těžko čitelný a dokonce ani slova nelze jednoznačně rozpoznat. [9]

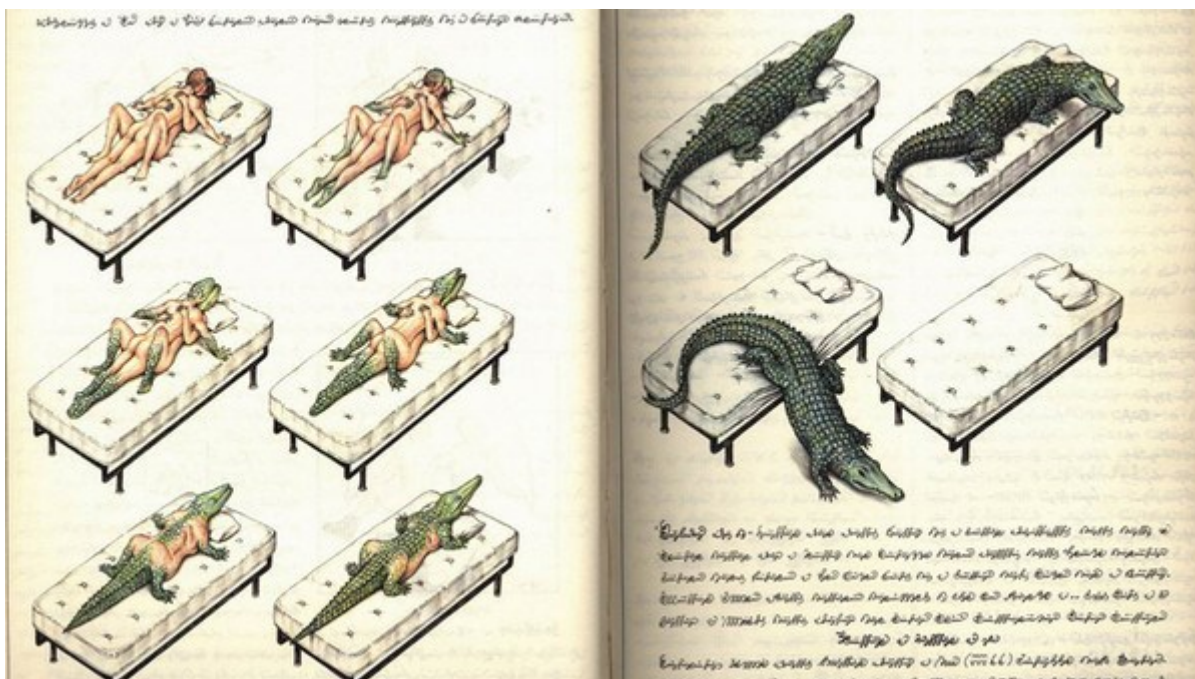


Obrázek 6: „Book of Soyga“ – ukázka jedné z 36 posledních stránek [9]

Jim Reed knihu detailně analyzoval a přišel na překvapující zjištění ohledně této hádanky. Každá tabulka je založena na „magickém slově“ o 6 znacích. Jednoduchá rovnice nám dovolí vypočítat všechny písmena mřížky 36 x 36 (viz. zdroj). [9]

### 2.5.2 Codex Seraphinianus

Ilustrovaná encyklopedie neexistujícího světa od italského umělce Luigiho Serafini. Obsahuje 300 stránek v šifrované abecedě a vymyšleném jazyce. Kniha je sama o sobě velmi prapodivná už jen ilustracemi. Pár mající sex se proměňuje v krokodýla (Obr. 7), rybí oči z nějakého divného stvoření apod. A tyto obrazy doprovází text, kterému nejde rozumět. [10]



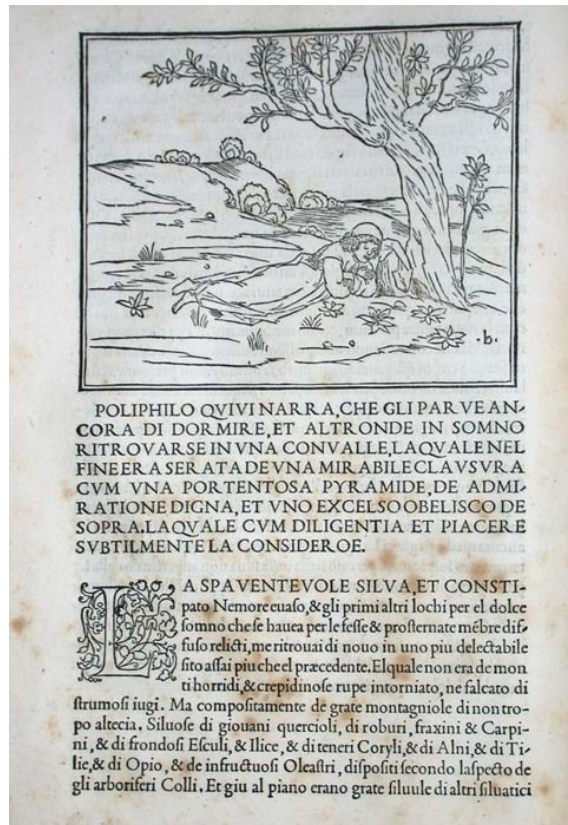
Obrázek 7: „Codex Seraphinianus“ ukázka z prapodivné knihy [10]

Představte si, že objevíte nový nepoznaný svět, ale bude odlišný od světa, co znáte, svět úplně naruby a budete chtít na tento svět „návod“, tak tohle kniha připomíná.

### 2.5.3 Hypnerotomachia Poliphili

Z řeckého *hýpnos*, „spánek“, *éros*, „láska“ a *machē*, „boj“. Kniha je psána latinou s italskou syntaxí a vypráví o Poliphilovi, který sní, skáče z jednoho snu do druhého a zažívá různá dobrodružství a úskalí a to vše kvůli jeho milované, Polii, která se mu vyhýbá.

Někteří vidí text jako stereotypní produkt své doby. Liane Lefavre, například naznačuje, že jde v mnoha ohledech o nepopsatelný příklad „vysoce stylizovaného žánru“. Profesor Weiss, mezitím prohlásil, že je kniha vážný zájemce o post nejnudnější práce v italské literatuře, avšak kniha má i své zastánce, přece by jinak nebyla mezi Top 10 nejzáhadnějšími rukopisy na světě. [11]



Obrázek 8: „Hypnerotomachia Poliphili“ ukázka ilustrace s textem [11]

## 2.5.4 The Oera Linda Book

Rukopis psaný v jazyce Old Frisian s neznámým autorem. Rukopis vyšel na světlo světa v roce 1867, přeložen do němčiny byl v roce 1872 a do angličtiny v roce 1876 s žhavou debatou o jeho autentičnosti. Většina, co knihu zkoumala, došla k závěru, že jde o falzum. Nicméně Herman Wirth byl přesvědčen o pravosti rukopisu a sepsal knihu roku 1933, v prvním roce Hitlerova režimu. Heinrich Himmler s Hermanem Wirthem souhlasil a to dalo vzniknout Himmlerově archeologické organizaci Ahnenerbe. [12]

Témata objevující se v rukopise se týkají katastrofizmu, nacionalismu, matriarchátu a mytologii celkově. Text, tvrdí, že Evropa a další země byly po většinu svého času ovládány a kontrolovány hrstkou vyvolených zasvěcených bohyní Freye. [12]

## 2.5.5 The Ripley Scrolls

George Ripley byl augustiniánský kanovník a vlivný alchymista známý pro svou práci Sloučeniny Alchymy, které věnoval Edwardu IV v roce 1471. Rukopis je psán ve verších a odhaluje „jak na kámen mudrců“, který dokáže proměnit jakýkoliv kov ve zlato, a propůjčuje nesmrtelnost vlastníkovy. Jeho objev byl velmi vážený ve světě alchymie a popularita svitků přispěla k oživení alchymistického učení v Anglii, a to zejména v 17. století po občanské válce. [13]

Ví se o existenci 23 svitků a knihovna v Bodleian pět z nich vlastní. Ostatní jsou roztroušeny po různých institucích po Anglii. [13]

### 2.5.6 The Smithfield Decretals

Tato práce je sbírka středověkého církevního práva, jehož účelem je nahradit všechny předchozí. Původně sestaven v roce 1230, rukopis byl ve skutečnosti dokončen roku 1300 v jižní Francii. [14]



Obrázek 9: „The Smithfield Decretals“ ukázka textu a ilustrace

Ve známé formě lze považovat za původní obsah rukopisu jen hrstku částí rukopisu, protože většina byla přidána asi o 40 let později, pravděpodobně také ve Francii. Majitel v této době pověřil hrstku umělců k osvětlení různých částí rukopisu a tak můžeme o autentičnosti rukopisu jen pochybovat. [14]

### 2.5.7 The Rohonc Codex

Kodex dostal své jméno podle města Rohonc, které se nachází v západním maďarsku, nyní Rechnitz, Rakousko. Kodex byl zde v držení až do roku 1838, kdy byl darován Maďarské akademii věd Gusztáv Batthyány. [15]

Zdá se být středověkého původu s ručně psanými piktogramy a černě kreslenými ilustracemi, což má svůj původ v maďarských dějinách. Mnoho akademiků věří, že je kodex falzum, protože muž jménem Sámuel Literáti Nemes, maďarský antikvář ve stejném časovém období přispíval akademii. [15]

I přes všechny závěry, že jde o falzum, se mnoho vědců a lingvistů pokoušelo rukopis rozluštit, avšak bez úspěchu. Kodex se skládá ze 448 stránek, obsahuje 87 kreseb, které obsahují náboženské vojenské scénérie. S kombinací náboženských symbolů se zdá, že se text má číst zprava doleva. [15]

Jazyk se zdá být příbuzný maďarštině, avšak obsah zůstává utajen. (Obr. 10)





Obrázek 10: „The Rohonc Codex“ ukázka textu s ilustrací

## 2.5.8 The Red Book

Rukopis svázaný v červené kůži vytvořený švýcarským psychiatrem Carl Gustavem Jungem v letech 1915 až asi 1930. [16]

Jung v knize zaznamenává své vize a fantazie, vypráví zde svůj osobní příběh a zákeřný sestup do toho, čemu mnozí říkali šílenství a poté jeho triumfální návrat jako nový člověk. Tento proces trval téměř 20 let. [16]

Styl psaní se podobá filozofovi Friedricha Nietzscheho v knize „Tak pravil Zarathustra“, Jung tuto knihu očividně četl a zůstal jí velmi ovlivněn. [16]



Obrázek 11: „The Red Book“ ukázka ilustrace knihy [16]

## 2.5.9 Prodigiorum ac ostentorum chronicon

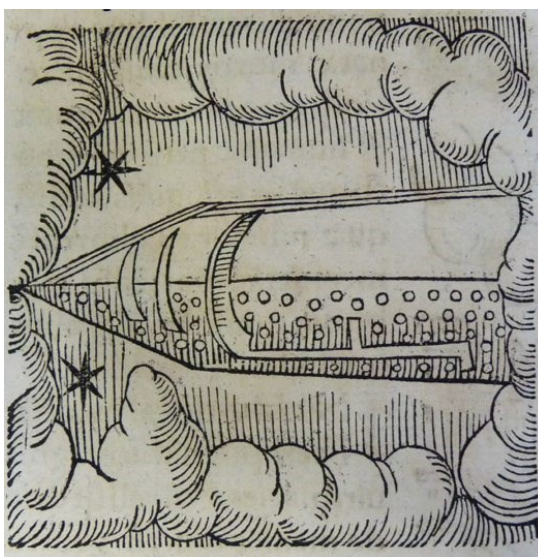
Jeden z nejslavnějších proroků západního světa je nepochybně Nostradamus, díky jeho proroctvím o biblickém konci světa apod. V roce 1557, kdy Nostradamus „tvořil“ své předpovědi, jiný vědec, Conrad Lycosthenes, publikoval rozsáhlý rozbor z historie podivných, nádherných a děsivých událostí a kretur od Adama a Evy až do jeho života. [17]

Tato „Kronika předzvěstí a proroctví“ vyšla v Basileji 1557. Je plná nádherných ilustrací, záhadných symbolů, monster a přírodních jevů. Patří sem i pozorování Halleyovy komety, invaze kobylek (Obr. 12), znetvořená zvířata atd. [17]



Obrázek 12: „*Prodigiorum ac ostentorum chronicon*“ invaze kobylek [17]

Nejslavnější obraz je kometa viděná nad Arábií v roce 1579 připomínající kosmickou loď, obraz velmi populární mezi fanoušky UFO. (Obr. 13) [17]



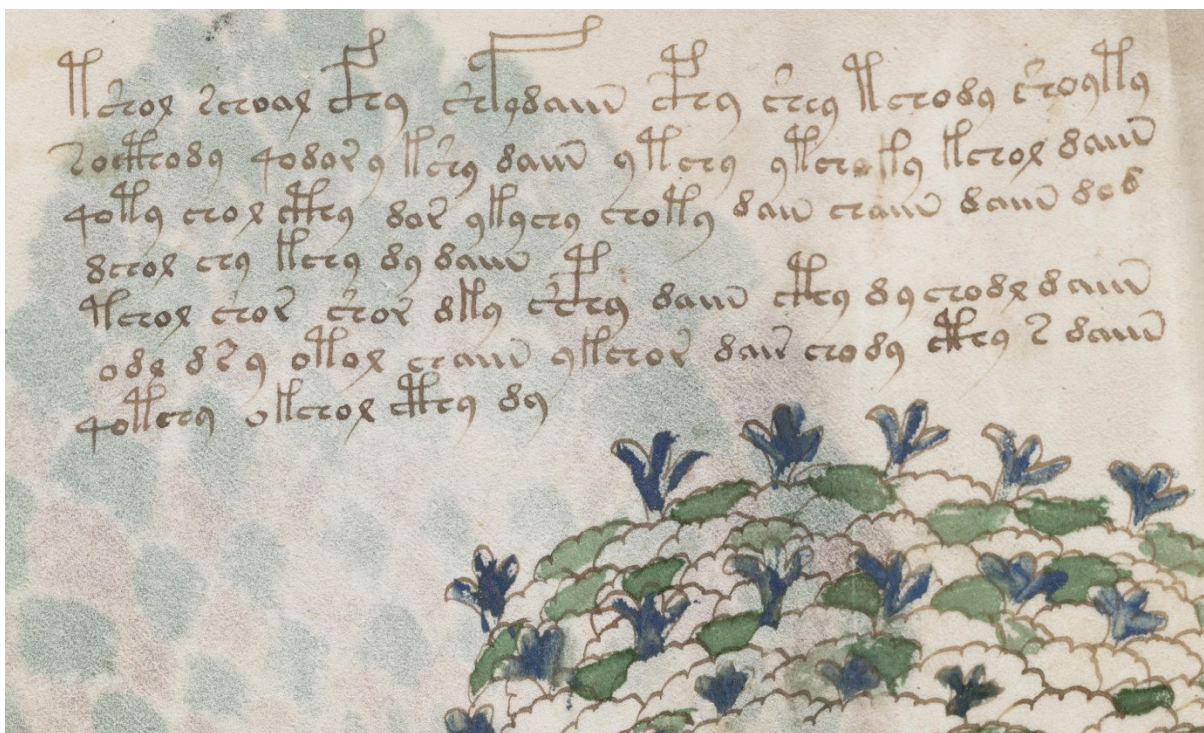
Obrázek 13: „*Prodigiorum ac ostentorum chronicon*“ kosmická loď

## 2.6 Voynichův rukopis

Voynichův rukopis, nejzáhadnější kniha světa. V padesátých letech minulého století se rukopis stal předmětem bádání u NSA. Dokonce ani vědci NSA nebyli schopni rukopis rozluštit. Jejich závěrem bylo, že rukopis používá polyalfabetickou substituční metodu, kdy se nějaké znaky zaměňují za jiné s použitím více abeced. O rukopise koluje mnoho mýtů a faktů, které se navzájem podporují či vyvrací. Jediný fakt, o kterém je možné říci, že je pravdivý, je snad jen období vzniku a to někdy mezi léty 1404 – 1438, určeno radiokarbonovou metodou datování. Nutno dodat, že i autor rukopisu je neznámý a existuje mnoho teorií i o tom kdo knihu napsal. Kniha dostala jméno podle svého objevitele polského původu Wilfrida M. Voynicha, který našel knihu roku 1912 v držení Jezuitským řádem.

Ohledně rukopisu existuje prakticky jen obrovské množství teorií a vlastně žádná z nich nemá jasné podklady. Už jen z tohoto důvodu je jakýkoliv výzkum týkající se knihy velmi náročný, jelikož nad většinou parametrů, které se používají k luštění šifer, či čtení neznámých písem, visí velký otazník a bez těchto chybějících kousků skládačky se zdá možnost, jak textu někdy v budoucnu porozumět, nepravděpodobná. Některé teorie dokonce hovoří o českém původu rukopisu a postup luštění rukopisu se vztahem k češtině by mohl být velmi zajímavý.

Ačkoliv se tedy rukopis považuje za šifrovaný, tak dodržuje Zipfův zákon, což je u šifer neobvyklé a většinou nemožné. Zipfův zákon je pravidlo matematické analýzy, které říká, že v textu psaném v přirozeném jazyce existuje konstantní rozdělení četnosti výskytu slov. Toto tvrzení tedy také odporuje závěru od NSA. Aby šifra dodržovala Zipfův zákon, musela by být vlastně dost jednoduchá, například substituční metoda s jednou abecedou, zde Zipfův zákon zůstane po šifrování dodržen.



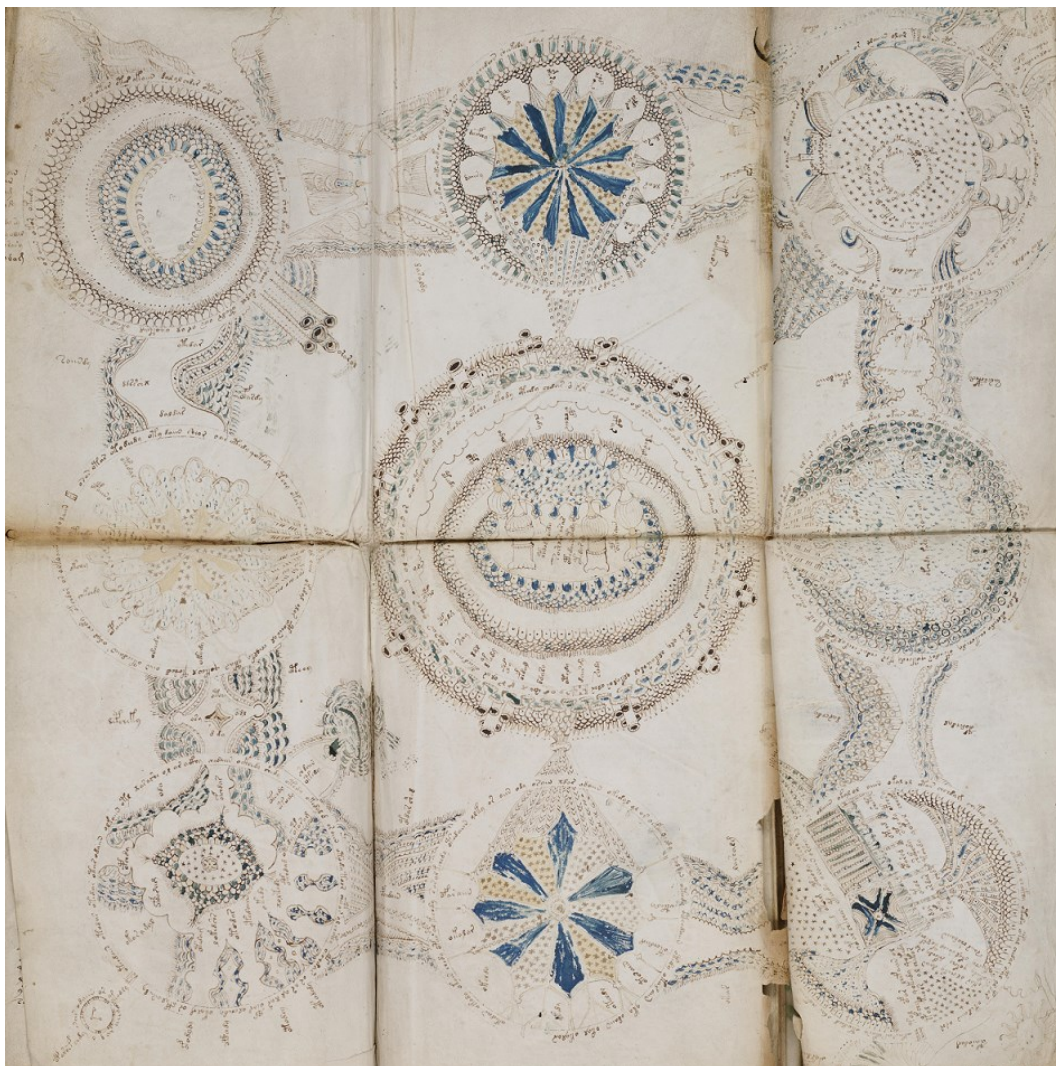
Obrázek 14: Ukázka písma z Voynichova rukopisu

## 2.6.1 Obsah

Říci něco o obsahu knihy, které nikdo nerozumí, ať už vlivem neznámého jazyka, či šifrováním, je poněkud bláhové. Knihu však doplňují pestré ilustrace, které mnohé napoví. Je možné rozpoznat kresby rostlin evropského původu, či dokonce kresbu připomínající slunečnici, která pochází z Ameriky, což je podivné, neboť byla objevena téměř o sto let později. Stále je ale vyobrazeno mnoho rostlin neznámého původu a nelze je identifikovat. Dále se vyskytují vyobrazení astronomická, náčrty zvěrokruhů atd. či náčrty a vyobrazení jakýchsi map připomínající náčrtky vesmíru apod.

Tímto se dá obsah rozdělit do několika částí:

- 1) Botanická
- 2) Astronomická (Obr. 15)
- 3) Biologická
- 4) Kosmologická
- 5) Farmakologická
- 6) Někaké druhy postupů



Obrázek 15: „Voynichův rukopis“ ilustrace z astronomické části rukopisu

## 2.6.2 Rukopis

Nejprve je třeba zmínit, že mnoho teorií tvrdí, že jde o text bez jakéhokoliv obsahu a smyslu a sloužil jen ke klamání v době vysoké nevzdělanosti. Písmo z části připomíná hebrejštinu a existuje teorie, která tvrdí, že je jen vizuálně kódovaná. Další teorie hovoří o: možnosti přirozeného avšak neznámého jazyka, známý jazyk avšak psán upraveným či vymyšleným písmem a nebože jde o mandžuštinu. Překlad titulní strany z tohoto jazyka připomínal jakýsi prolog a úvod, ale text byl jen těžko srozumitelný.

Abeceda „Voyničiny“ obsahuje 31 znaků viz. tabulka v příloze, na 240 stranách (původně asi 272) v 17 arších o zhruba 35 000 slovech psaných na pergamenu.

Nabízí se možnost zašifrovaného a navíc neznámého jazyka, a tak jediné spojení s možností porozumění textu jsou ilustrace, které mohou napomoci porozumění textu týkajícího se přímo těchto kreseb. Osobně si myslím, že se jedná o tento případ, takže vlastně jakýkoliv pokus o rozšifrování může vést tímto zase k nečitelné podobě.

## 2.6.3 Přepis

Existuje mnoho různých přepisů do počítačově čitelné formy (Bennet, Currier), pro své řešení programu určeného k analýze jsem si vybral přepis od FSG Williama Friedmana kupodivu z roku 1940, protože už od prvního pohledu na přepis se mi zdál dobře řešený a vyhovující pro mou práci.

## 3 Praktická část

V rámci praktické aplikace jednoduché analýzy přepisu Voynichova rukopisu se budu nejprve věnovat vysvětlení důvodu druhu použité platformy. Dále problematice čtení FSG přepisu a jeho zpracování, protože se stala podstatnou částí celé práce a na jaké problémy jsem narazil a také samozřejmě vysvětlím jejich řešení. Vybral jsem si jazyk C# na platformě .NET s použitím Windows Forms.

Rád bych zdůraznil, že veškeré zde zobrazené ukázky kódu slouží výhradně jako nástin pro přehled jak implementace funguje, veškeré mezery v pochopení kódu je možné si doplnit jeho přečtením s důkladným komentováním.

### 3.1 Architektura řešení

Implementaci jsem rozdělil do několika komponent (knihoven) tak, aby od sebe byly odděleny části podle řešení. Název programu je `AnalysisUnknownTexts` a skládá se ze dvou knihoven a části starající se o Windows Forms:

- **Transcript** (Class Library)
- **Statistics** (Class Library)
- **AnalysisUnknownTexts** (WinForms)

**Transcript** je knihovna, ve které se nachází kód doménového modelu, který se stará o vše okolo načtení textového souboru přepisu dle definovaných pravidel (konstant). Doménový model slouží ke složení knihy do čitelné podoby pro zbytek programu.

**Statistics** je knihovna sloužící k provedení základní analýzy nad sestavenou knihou. Analýzu typu počet písmen, slov a sousloví. Knihovna je připravena k rozšíření v pozdějším stádiu výzkumu v rámci fraktálů.

**AnalysisUnknownTexts** je část programu starající se o vizuální stránku tzn. GUI, diagramatické části programu apod. Obsahuje také jednodušší funkce týkající se filtrování zobrazeného obsahu nebo jeho uspořádání a také zobrazení dat v celém jejich průběhu zpracování. Také se stará o vizualizaci grafů a „mapy“ knihy.

Přímo kód programu je detailně okomentovaný, důkladně v něm popisují veškeré náležitosti a přesně co provádí. Pro komentáře je použit český jazyk.

### 3.2 Knihovna Transcript

V této podkapitole se věnuji detailní struktuře kódu starajícího se o samotný přepis. Všechny problémy, na které jsem narazil a jejich řešení.

Knihovna obsahuje 4 třídy:

- 1) Cell
- 2) Word

- 3) Page
- 4) Book

Třída Book dále obsahuje podtřídu Constants, která definuje pravidla pro čtení přepisu. (oddělovač slov, řádku či odstavců apod.)

### 3.3.1 Třída Constants

Název sice mluví sám za sebe, ale ve zkratce se jedná o statickou třídu, která definuje pravidla pro čtení přepisu. Určuje, kterými znaky v přepise se definuje oddělení slov či odstavců apod.

```
// Rozlišovací znak pro popis (s mezerou za) a pro komentáře
public const char DescCommSep = '#';

// Oddělovací znak pro odstavce
public const char ParaSep = '=';

// Oddělovací znak pro řádky
public const char LineSep = '-';

// Oddělovací znak pro buňky slov
public const char CellSep = ',';

// Stanovení znaku používaného pro generování všech kombinací nejasných slov
public const char TemplateWorkMark = '.';

// Rozlišovací znak pro začátek nejasnosti ve slově
public const char NotClearBracketLeft = '(';

// Rozlišovací znak pro konec nejasnosti ve slově
public const char NotClearBracketRight = ')';

// Rozlišovací znak pro oddělení možností nejasnosti ve slově
public const char PipeSep = '|';

// Nahrazovací znak pro nečitelné znaky z rukopisu
public const char MissLettSep = '_';

// Vzorec pro rozdělení buněk jen podle oddělovače mimo nejasná slova
public const string CellSepPattern = @"(?:[^(,]|\\([^\)]*\))+";

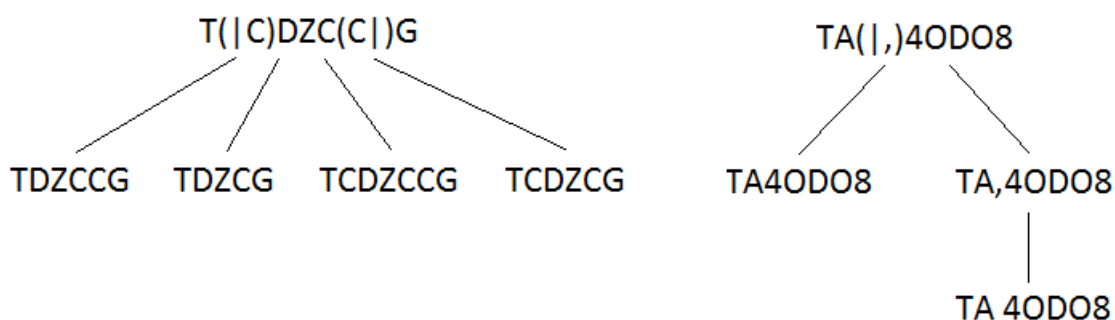
// Vzorec pro hledání možností v nejasných slovech
public const string NotClearBracktesPattern = @"\..*?\\";
```

Předposlední je vzorec používaný k nalezení čárky, jako oddělovače slov, která je uzavřena v závorce a tímto v kódu docílit jejího vynechání (nemůžeme si být jistí, zda jí použít nebo ne). Poslední slouží k nalezení celé závorky, která obsahuje nejasnosti (použití s generováním kombinací v buňce).

### 3.3.2 Třída Cell

V přepisu, který byl konkrétně v našem řešení použit, je občas možné narazit na slova, která nejsou úplně jasná (Při vytváření přepisu z rukopisu například nebylo možné jasně přečíst slovo, či určit mezeru mezi slovy apod.) Kvůli tomu jsem tedy vytvořil třídu, která reprezentuje jednoduše „buňku“

(reprezentace jednoho nebo více slov), protože vytvoření kombinací nejasných slov občas může poskládat slov více, více na Obr. 16.



Obrázek 16: Ukázka výskytu možných nejasností v přepisu

Při čtení a sestavování knihy v programu umí kód takovéto nejasnosti detekovat a adekvátně se zachovat. Implementace třídy dále obsahuje vlastnosti jako číslo řádku, na kterém se buňka vyskytuje, číslo neboli ID buňky v rámci celého přepisu apod. Vlastnost Word typu List<string> obsahuje výčet všech již jasně určených slov podle jejich váhy. Vlastnost Content typu Dictionary (slovník) s T parametry string v roli klíče a int v roli hodnoty obsahuje všechny vygenerované kombinace nejasných slov. Pokud je slovo jasně určeno z textu, obsahem těchto vlastností je to slovo! Ale pokud slovo jasné není, tak nejdůležitější částí kódu této třídy je metoda Init, která obsahuje dvě větve. Ta první jednoduše přiřadí jasné slovo, avšak ta druhá provádí generování kombinací viz. ukázka kódu.

Ukázka kódu generování těchto kombinací:

```

string _input = this.Origin;

Regex _regex = new Regex(Constants.NotClearBracketsPattern);

string _template = _regex.Replace(_input, Constants.TemplateWorkMark.ToString());

string[] _matches = _regex.Matches(_input).Cast<Match>().Select(x =>
    x.Value).ToArray();

List<string[]> _extract = new List<string[]>();

for (int i = 0; i < _matches.Length; i++)
{
    _extract.Add(_matches[i].Substring(1, _matches[i].Length - 2).Split(new char[] {
        Constants.PipeSep }));
}

this.GenerateCombinations(_template, "", _extract, 0);
  
```

Kód si prakticky jen připraví „půdu“ pro generování kombinací. Ze začátku se definuje regulární výraz, poté se dle něj vytvoří šablona, kam se budou možnosti vkládat a vyjmou se veškeré nejasné části slova. Následně se projdou všechny tyto nejasné části a vytvoří se z nich List všech nejasností, kde každý prvek bude obsahovat všechny možnosti této „nejasnosti“. Pro upřesnění, když si vezmeme levé slovo z Obr. 16, tak v tomto případě bude List velikosti 2, protože slovo má dvě



nejasnosti a poté pole bude obsahovat všechny možnosti, pro List indexu 0, to bude prázdný znak a znak „C“. Nakonec se zavolá rekurzivní metoda, která naplní vlastnost Content všemi kombinacemi.

Metoda generování kombinací:

```
private void GenerateCombinations(string template, string temp, List<string[]>
    extract, int counter)
{
    if (counter == extract.Count)
    {
        string _word = template;
        string[] _temp = temp.Split(new char[] { Constants.PipeSep });

        for (int i = 0; i < _temp.Length - 1; i++)
        {
            int _pos = _word.IndexOf(Constants.TemplateWorkMark);

            _word = _word.Substring(0, _pos) + _temp[i] + _word.Substring(_pos
                + 1);

            if (!this.Content.ContainsKey(_word))
            {
                this.Content.Add(_word, 0);
            }
        }
    }
    else
    {
        foreach (var item in extract[counter])
        {
            this.GenerateCombinations(template, temp + item.ToString() +
                Constants.PipeSep, extract, counter + 1);
        }
    }
}
```

Kód je ve své podstatě velmi jednoduchý. Jako vstup potřebujeme znát šablonu, do které vkládáme kombinace nejasností, temp který skládá vícero nejasností dohromady, List nejasností a nakonec počítadlo průchodů. Kód následně při dovršení průchodu všech nejasností, tedy průchod všech prvků Listu vloží tyto nejasnosti do šablony, jinak prochází v rámci listu a skládá kombinace.

### 3.3.3 Třída Word

Tato třída oproti třídě Cell už obsahuje jasně definovaná slova v knížce, je tedy sestavena průchodem všech buněk knihy a průchodem všech slov v buňce.

Metoda z třídy Book, která toto sestavení provádí:

```
private void BuildBook()
{
    int _id = 0;
    this.Word = new List<Word>();
    foreach (var item in this.Cell)
    {
```

```

        foreach (var item2 in item.Word)
        {
            this.Word.Add(new Word(_id, item.Id, item2));
            _id++;
        }
    }
}

```

Toto roztrídění celkově zjednodušuje práci s daty z přepisu pro analýzu a hlavně jejich zobrazení.

### 3.3.4 Třída Page

Třída reprezentující stránku z knihy (přepisu). Uchovává údaje při čtení přepisu o stránkách, jejich komentářích a popisu. V implementaci programu nehraje skoro žádnou roli a je určena pro budoucí použití.

### 3.3.5 Třída Book

Dovolím si říci, že se jedná asi o nesložitější třídu z celého programu. V rámci vytvoření instance Book zahrnuje jak čtení knihy, tak její parsování podle odstavců, řádku a slov. Následně sestaví slova bez jakékoliv úpravy. Dále je obsažena velmi důležitá metoda starající se o udělení vah kombinacím nejasných slov podle výskytu v textu.

Třída obsahuje tři vlastnosti typu List a to Page, Cell a Word, s adekvátními T parametry:

```

public List<Page> Page { get; private set; }
public List<Cell> Cell { get; private set; }
public List<Word> Word { get; private set; }

```

V rámci konstruktoru vytvoříme jejich instance. Třída Book obsahuje metodu Init, která je volána zvlášť a to s parametrem v podobně cesty k textovému souboru (neboť se cesta definuje až za běhu programu). Dále následuje kód, který obstarává samotné čtení ze souboru.

Nejprve je soubor otevřen pomocí statické třídy a její metody, která je součástí C#, File.OpenText a obsah souboru vložen do pole stringů pomocí metody ReadToEnd. Následně se prochází jednotlivé řádky souboru a rozlišují se popisy, jednotlivé stránky a komentáře. Myslím si, že tuhle část není třeba celou rozebírat, čtenář může nahlédnout do okomentovaného kódu v příloze.

Zaměřím se na parsování samotného textu jednotlivých stránek:

```

string[] _para = _text.Split(new char[] { Constants.ParaSep },
    StringSplitOptions.RemoveEmptyEntries);

for (int j = 0; j < _para.Length; j++)
{
    string[] _line = _para[j].Split(new char[] { Constants.LineSep },
        StringSplitOptions.RemoveEmptyEntries);

    for (int k = 0; k < _line.Length; k++)
    {

```

```

string[] _cell = new Regex(Constants.CellSepPattern)
    .Matches(_line[k]).Cast<Match>().Select(m => m.Value).ToArray();

for (int l = 0; l < _cell.Length; l++)
{
    int _lineInTranscript = i;

    _paraInPageCounter = j; // Počítadlo odstavců dle j
    _lineInPageCounter = k; // počítadlo řádků dle k
    _cellInLineCounter = l; // Počítadlo buněk dle l

    this.Cell.Add(new Cell(_cellCounter, _cell[l],
        this.IsClear(_cell[l]), _lineInTranscript, _pageCounter,
        _lineCounter, _paraInPageCounter, _lineInPageCounter,
        _cellInLineCounter));

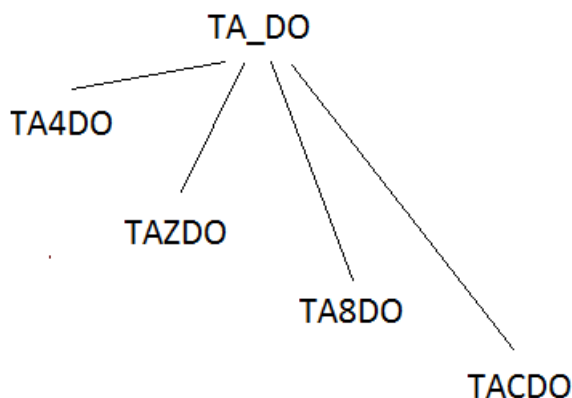
    _cellCounter++; // Počítadlo buněk celkově
}

_lineCounter++; // Počítadlo řádků celkově
}
}

```

V rámci tohoto kódu se text stránky nejdříve dělí dle odstavců a následně dle řádků a slov. Slova se dělí samozřejmě jen podle oddělovače slov, který je od pohledu již jasný. Nakonec se vytvoří instance třídy Cell a přidá se do Listu. Na konci čtení se volá funkce BuildBook v rámci zobrazení „mezivýsledků“.

Další složitější metoda (ClarifyBook) se stará o spočítání váhy jednotlivých nejasných slov. Prochází se všechny buňky, které jsou definované jako „nejasné“. Jelikož se v přepisu mohou vyskytovat i slova s chybějícími znaky, tak má tahle metoda něco jako „před-krok“, kdy se projde celý seznam jasných slov a najdou se veškeré možnosti a kombinace s doplněním znaků. (viz. Obr. 17)



Obrázek 17: Možnosti jak může slovo vypadat s porovnáním vůči textu

Pro vyhledávání slov v knize se zde použije vzor TA\_DO a díky průchodu textem se v tomto případě najdou 4 varianty tohoto slova, všechny se do obsahu buňky přidají a původní nejasné slovo se smaže, viz. ukázka kódu:

```

Regex _regex = new Regex(@"\" + item2.Replace(Constants.MissLetSep,
    Constants.TemplateWorkMark));

foreach (var item3 in this.Cell)
{
    foreach (var item4 in item3.Word)
    {
        if (_regex.IsMatch(item4) && item2 != item4)
        {
            _wasFind = true; // Nastavení detekce na "nalezeno"

            if (!item.Content.ContainsKey(item4))
            {
                item.Content.Add(item4, 0);
            }
        }
    }
}

```

Kód běží v cyklu procházejícím všechny kombinace slov, kterým chybí nějaké písmeno. Jakmile se provede tato operace nad všemi slovy s chybějícími znaky, může program přikročit k jádru metody, což je vypočtení vah jednotlivých kombinací nejasných slov, viz. ukázka kódu:

```

int _counter = 0; // Počítadlo

string[] _word = item2.Split(new char[] { Constants.CellSep });

for (int i = 0; i < _word.Length; i++)
{
    _counter += this.WordCount(_word[i]);
}

if (item.Content.ContainsKey(item2))
{
    item.Content[item2] = _counter;
}
else
{
    item.Content.Add(item2, _counter);
}

```

Kód opět běží v cyklu procházejícím všechny kombinace slov. Pro každou kombinaci se vypočítá váha, tedy počet výskytů v celém textu (v rámci více slov na buňku se počítá se sumou, počítám s tím, že pro další výzkum bude nutné použít složitější výpočet váhy u více slov).

Nakonec se slovník seřadí dle vah a první se použije pro reprezentaci slov(a) v buňce a zavolá metoda BuildBook, která znovu poskládá slova knihy, viz. kód:

```

item.Content = item.Content.OrderByDescending(x => x.Value)
    .ToDictionary(x => x.Key, x => x.Value);

item.Word = item.Content.FirstOrDefault().Key
    .Split(new char[] { Constants.CellSep }).ToList();

```

## 3.4 Knihovna Statistics

Knihovna obsahuje kód starající se o provedení analýzy nad textem získaným knihovnou Transcript.

Knihovna obsahuje pouze 2 třídy:

- 1) AnalyzedWord
- 2) Analysis

### 3.4.1 Třída AnalyzedWord

Třída se stará o uchování všech analyzovaných slov. V rámci třídy deklaruji vlastnost Word, která je Listem a obsahuje všechna umístění již analyzovaných slov v rámci jedné shody a vlastnosti WordCount, která uchovává počet slov v analyzovaném sousloví.

Konstruktor se stará o vytvoření instance vlastnosti Word a zároveň předává první analyzované slovo v rámci jednoho řetězce a počet slov v sousloví. Třída dále deklaruje funkci AddWord, kdy přidá nové umístění do Listu, funkci CompareTo z rozhraní IComparable a přepisuje funkci ToString, aby vracela počet slov.

Ukázka konstruktoru:

```
public AnalyzedWord(Word word, int wordCount)
{
    this.Word = new List<Word>();
    this.Word.Add(word);
    this.WordCount = wordCount;
}
```

### 3.4.2 Třída Analysis

Zde implementuji dvě metody, kde první se stará o jednoduché sečtení všech unikátních symbolů (znaků) vyskytujících se v textu přepisu a druhá dělá to samé, ale se slovy a slovními spojeními. Obě metody využívají jako návratovou hodnotu Dictionary (slovník) s T parametry string (řetězec) v podobě klíče a AnalyzedWord v podobě hodnoty. Vstupní proměnou pak je instance třídy Book z knihovny Transcript, která odpovídá sestavené knize. U metody analyzující slova a jejich spojení je nutné ještě určit počet slov ve slovním spojení.

Na začátku obou metod je třeba vytvořit instanci návratové hodnotě. A poté jednoduchými cykly procházet slova knihy. U metody řešící znaky poté ještě jednotlivé znaky slov a u metody řešící i slovní spojení poté musíme vždy v každém porovnávání vytvořit *porovnávací řetězec* i několika slov (dle počtu slov ve slovním spojení viz. ukázka kódu).

Ukázka kódu sestavování řetězce pro slovní spojení:

```
string _compString = book.Word[i].Value;

if (wordCount > 1)
{
    for (int j = 1; j < wordCount; j++)
```

```

    {
        _compString += " " + book.Word[i + j].Value;
    }
}

```

Zde je kód probíhající v iteraci mezi jednotlivými slovy. Nejdříve definujeme lokální proměnou, do které uložíme prvotní hodnotu (první slovo). Následně, pokud je požadavek na slovní spojení, nezávislým cyklem projedeme dalších x slov a takto vytvoříme porovnávací řetězec.

Ukázka společné části kódu pro obě metody:

```

if (!_statistic.ContainsKey(_compString))
{
    _statistic.Add(_compString, new AnalyzedWord(book.Word[i], wordCount));
}
else
{
    _statistic[_compString].Add(book.Word[i]);
}

```

V této části se řeší unikátnost klíčů ve slovníku (klíč může být jedinečně unikátní). Avšak stejně je potřeba zjistit zda byl již klíč nalezen či nikoliv abychom do již existujícího přidali další výskyt slova nebo písmena.

## 3.5 GUI (WinForms) – AnalysisUnknownTexts

Součástí této podkapitoly je rozbor formulářů Windows Forms užitých k ovládání aplikace, rozboru dat a vizualizaci dosažených výsledků.

### 3.5.1 Formulář MainForm

MainForm je hlavní část zobrazující veškeré ovládání aplikace (programu). Načtení adresy souboru, spuštění různých funkcí apod. a samozřejmostí je zobrazení výsledků a mezivýsledků.

Formulář obsahuje StatusBar zobrazující textovou informaci o aktuálně prováděné činnosti. Kód formuláře obsahuje velmi triviální metody v podobě událostí jako klik na tlačítko apod. Pod těmito událostmi se poté volají jednotlivé většinou již popsání funkce, jako načtení souboru, vyjasnění textu, jednoduché zobrazení detailu tabulky, nebo evokování spuštění analýzy při pohybu prvním TrackBarem zleva, kdy první slouží k definování, zda se jedná o písmeno, slovo, anebo počet slov. A druhý TrackBar definuje minimální počet výskytů. Dále zobrazení histogramů a detailu jednotlivých záznamů v tabulkách atd.

### 3.5.2 Formulář DetailCellForm

Formulář slouží k zobrazení detailu buňky a veškerých informací o ní, včetně vygenerovaných kombinací u nejasných slov a zobrazení již vyjasněných.

Evokuje se při dvojkliku na záznam levé tabulky hlavního formuláře a kód formuláře je velmi triviální.

### 3.5.3 Formulář DetailAnalyzedWordForm

Podobně jako u předchozího, jen se jedná o analyzované slovo s odkazy na všechny výskyty v textu (je možné vyvolat Formulář DetailCellForm). Obsahuje funkcionalitu na vygenerování mapy celého přepisu s vyznačením analyzovaného písmena, slova nebo slov. Navíc ještě obsahuje CheckBox, který přepíná mezi dvěma módy zobrazení mapy.

Evokuje se při dvojkliku na záznam pravé tabulky hlavního formuláře a kód formuláře je opět velmi triviální, avšak obsahuje také kód pro generování obrázku s mapou knihy.

Při generování obrázku se sestavená kniha prochází podle řádků a sloupců, kdy se vykresluje co čtverec to slovo, s tím, že pokud na vykreslené místo odkazuje analyzované slovo, tak se čtverec viditelně obarví červeně.

### 3.5.4 Formulář BookMapForm

Velmi triviální formulář zobrazující mapu knihy. Do konstruktoru se předává Bitmapa a ta se zobrazí v PictureBox. Ten musí být obalen panelem s nastavenou pozicí na „Dock“ a zapnutým „autoscroll“, aby bylo možné Bitmapou scrollovat.

Evokuje se z formuláře o detailu analyzovaného slova.

### 3.5.5 Formulář HistogramForm

Opět velmi triviální formulář zobrazující histogram analyzovaných dat, Do konstruktoru se předají data, která chceme zobrazit a poté se pomocí cyklu naimportují do C# třídy „Chart“.

Evokuje se z hlavního formuláře.

### 3.5.6 Formulář DataGridViewForm

Formulář starající se o vytvoření kopie tabulky z hlavního formuláře k porovnání dat nebo mezivýsledků.

Evokuje se tlačítkem pod tabulkou na hlavním formuláři.

## 3.6 Použití aplikace (programu)

Stručný návod týkající se ovládání implementovaného programu, neboť mnohé funkce mohou zůstat nepovšimnuté, pokusím se zde vysvětlit postup v několika krocích.

### 3.6.1 Hlavní formulář (MainForm)

- Po spuštění programu se zobrazí okno (viz. Obr. 18)
- Tlačítko „Load“ slouží k výběru textového souboru
- Tlačítkem „Clearify“ se spustí algoritmus, který vyčistí knihu od nepřesností ve slovech (nutno spustit 2x)

- Posunutím „Trackbarů“ napravo se evokuje funkcionalita analýzy
- Dvojklikem na prvek v seznamu tabulky zobrazíme jeho detail
- Tlačítkem „Show Detail“ pod tabulkami zobrazíme jejich náhled v novém okně
- Tlačítkem „Histogram“ se zobrazí histogram právě analyzovaných dat v tabulce napravo
- Tlačítkem „Without order“ vedle „Histogram“ se zobrazí neseřazený histogram právě analyzovaných dat v tabulce napravo

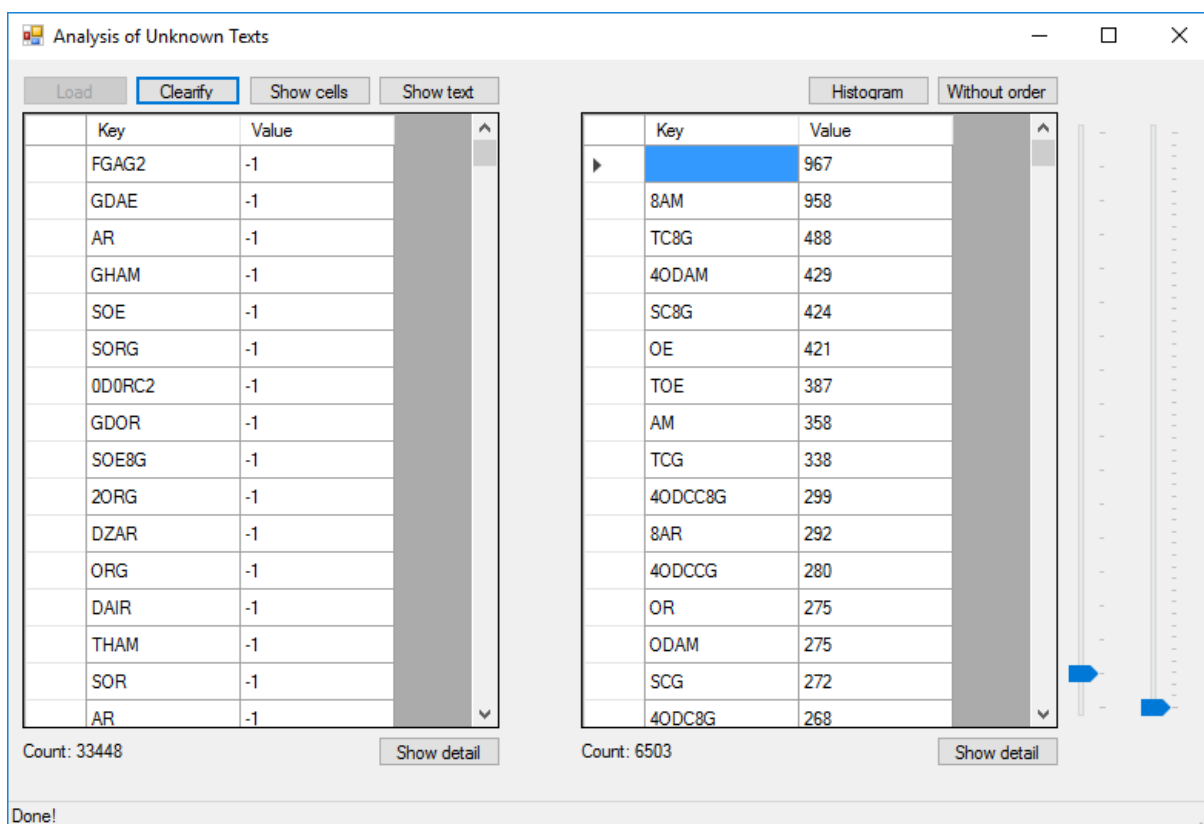
### 3.6.2 Formulář detailu analyzovaných dat (DetailAnalyzedWordForm)

- Tlačítkem „Show BookMap“ se zobrazí mapa průchodu knihou s vyznačením analyzovaných slov
- Změna stavu „CheckBoxu“ ovlivní zobrazení mapy

## 3.7 Ukázka aplikace (programu)

Pro lepší orientaci v aplikaci zde ukáží možné použití programu a případný průběh analýzy v několika obrázcích.

### 3.7.1 Hlavní formulář (MainForm)



Obrázek 18: Ukázka aplikace – hlavní formulář

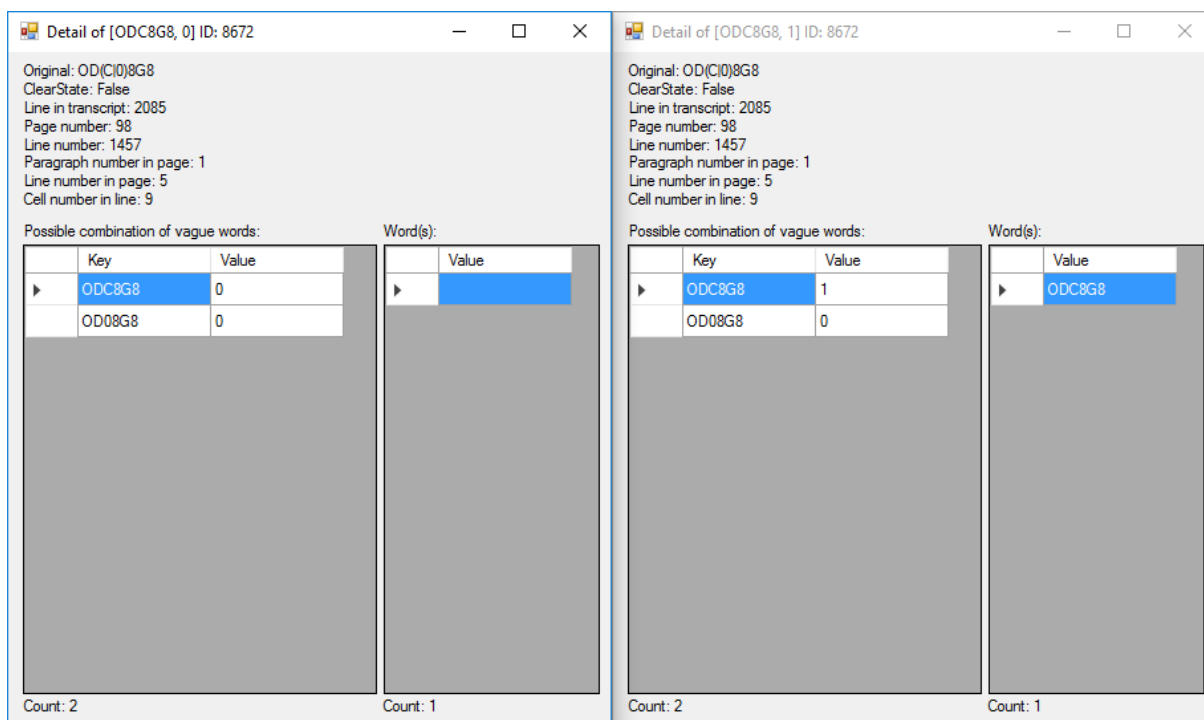


V levé tabulce je možné si u slov všimnout hodnoty „-1“, ta značí, že dané slovo bylo v textu jasně definované, při „0“, která značí váhu slova, se pak generovaly kombinace nejasností. Při kliku na tlačítko *Clarify* se určí váhy jednotlivých kombinací (akci je nutné provést 2x).

Poté je možné si všimnout, že v pravé tabulce zmizí první prázdný záznam, který reprezentoval tato nejasná slova, zmizí. Tzn., že se všechna slova správně určila.

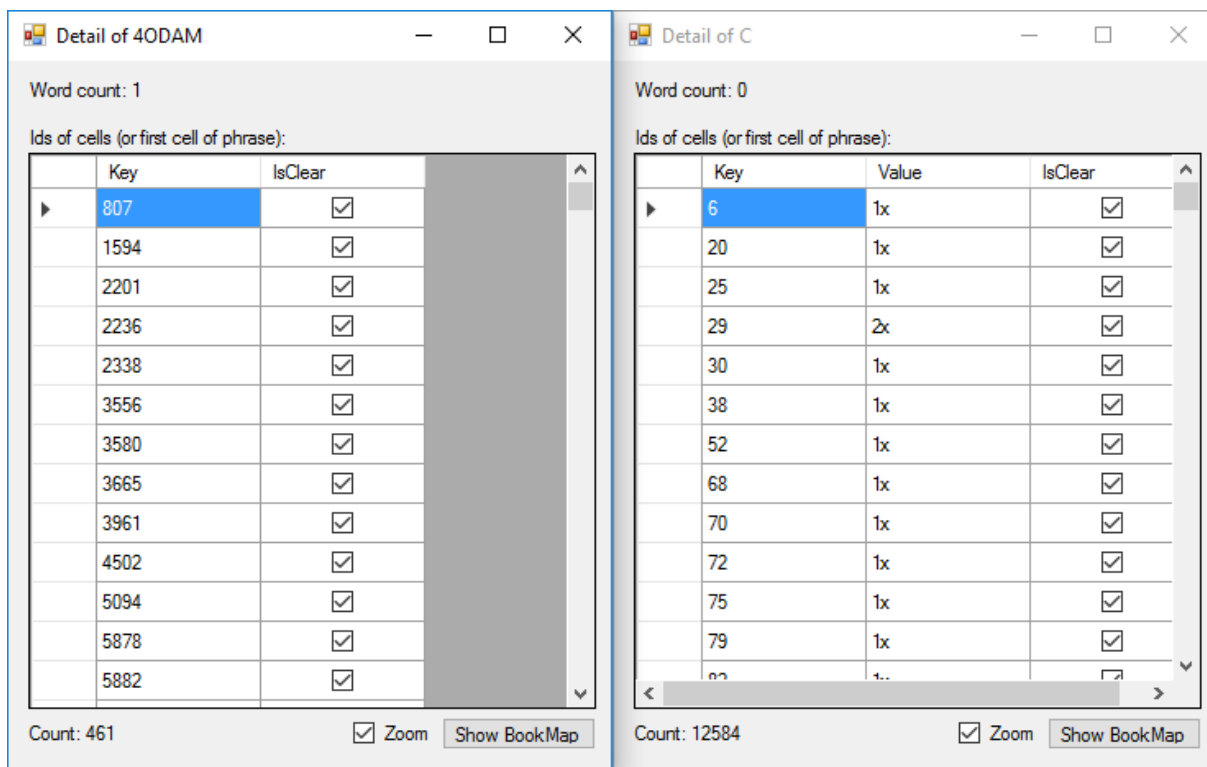
### 3.7.2 Formulář detailu buněk (DetailCellForm)

Jak již bylo řečeno, dvojklik na záznam v levé tabulce hlavní formuláře evokuje formulář na obr. 19. Tato ukázka přímo zobrazuje nejasné slovo OD(C|0)8G8 a jeho vygenerované reprezentace.



Obrázek 19: Ukázka aplikace – detail buňky před vyjasněním textu - vlevo a po - vpravo

### 3.7.3 Formulář detailu analyzovaných dat (DetailAnalyzedWordForm)



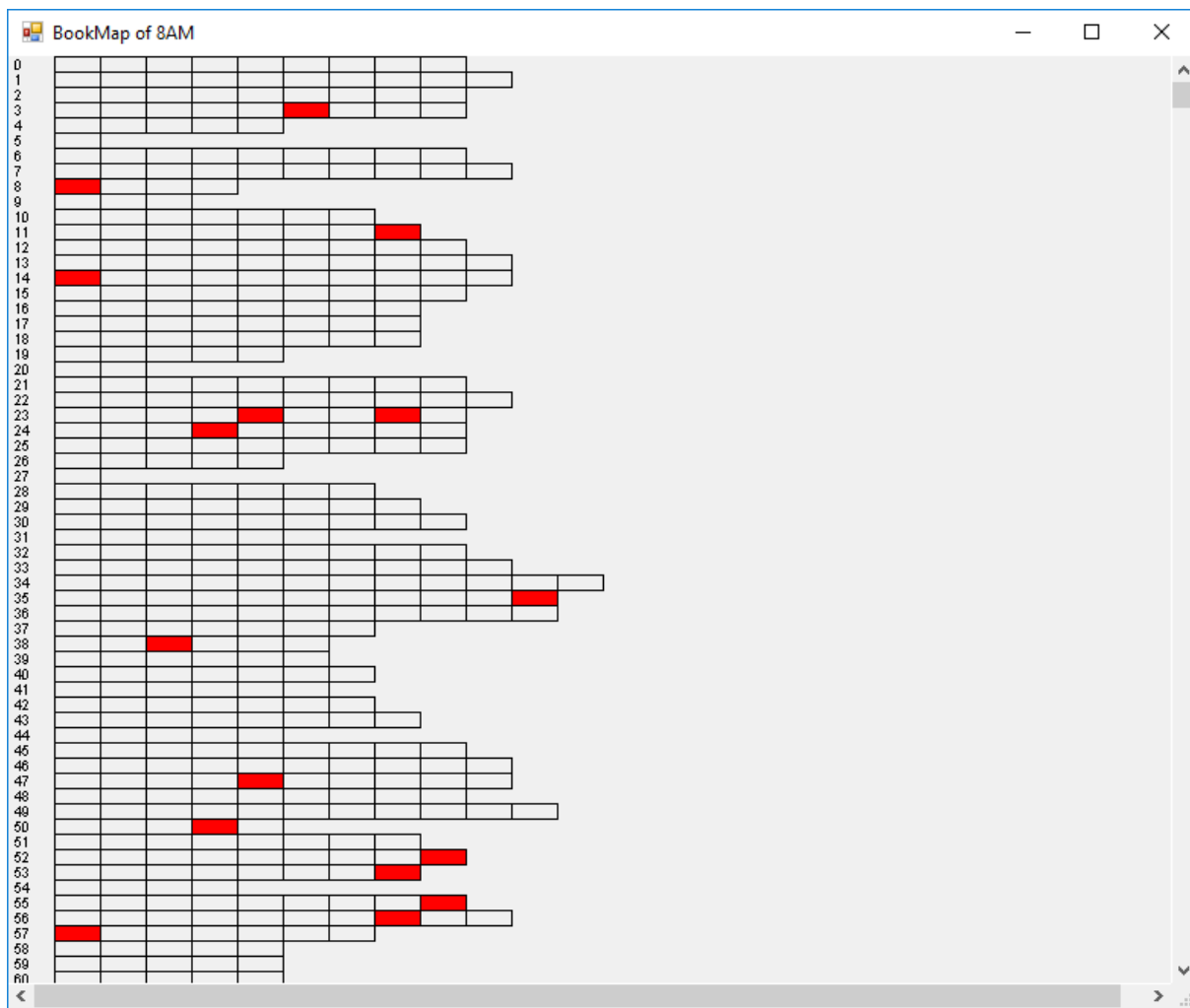
Obrázek 20: Ukázka aplikace – analyzované slovo - vlevo a znak - vpravo

Po dvojkliku na záznam pravé tabulky hlavního formuláře vyvolá detail analýzy. Vlevo je možné vidět detail analyzovaného slova nebo sousloví, v tomto případě slova (Word count: 1). Dále tabulka ukazuje, zda bylo slovo v textu jasně definované či nikoliv. Vpravo pak detail znaku, kdy se i objevuje počet výskytů ve slově. Dvojklik znovu odkazuje na detail jednotlivé buňky.

## 3.8 Výsledky

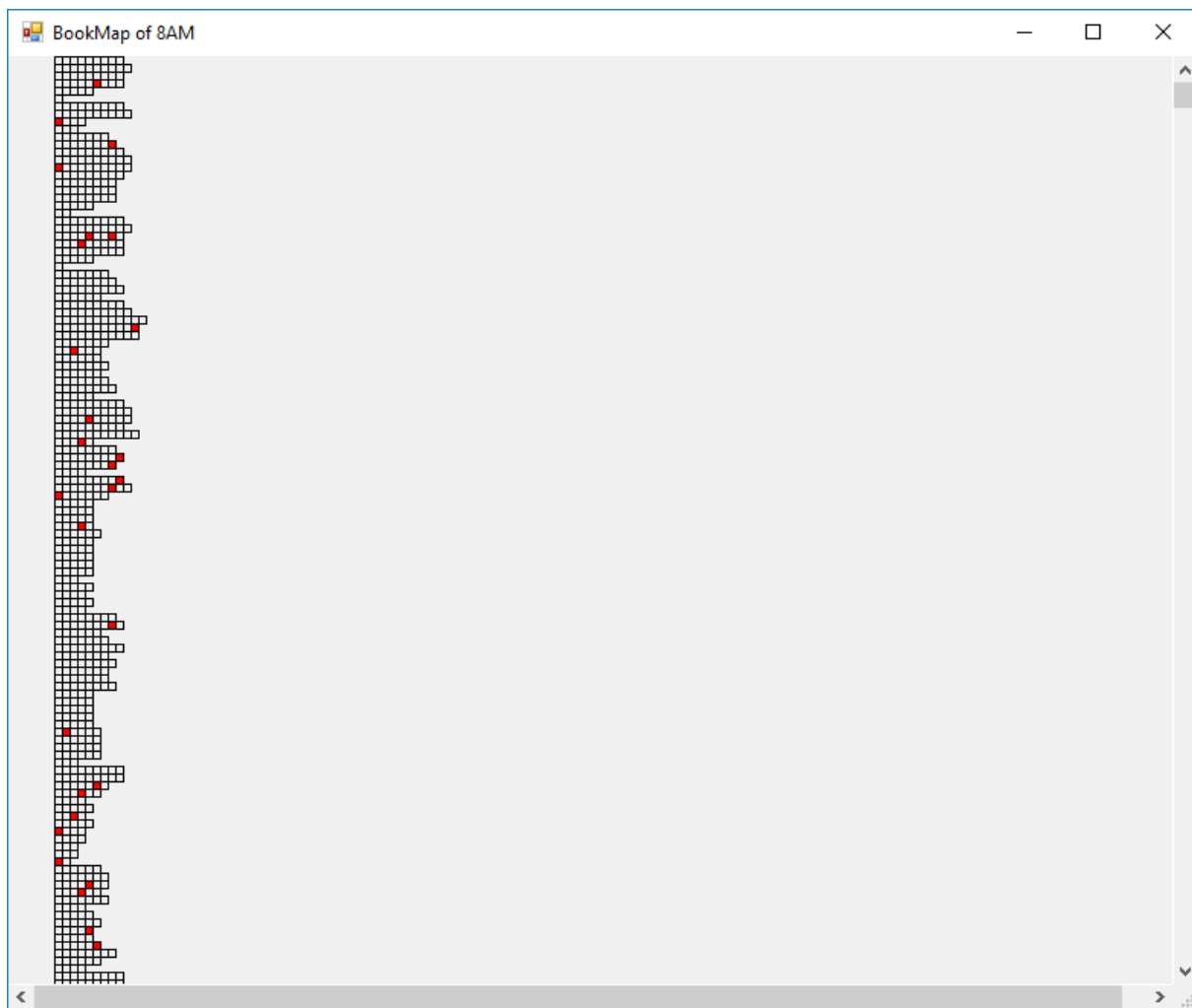
V předchozích dvou podkapitolách jsem předvedl, jak se aplikace používá, a jak průchod aplikací a analýza vypadá v praxi. Tato podkapitola zahrnuje ty části programu, které zobrazují, již samotné výsledky, či výstup programu vhodný pro další postup.





Obrázek 22: Výsledky – hlavní výstup aplikace - Mapa knihy (zvětšení)

Na Obr. 22 je možné vidět výskyt slova „8AM“ v rámci celého textu. Další krok vývoje bude úprava tohoto výstupu pro potřeby analýzy fraktály.



Obrázek 23: Výsledky – hlavní výstup aplikace - Mapa knihy (bez zvětšení)

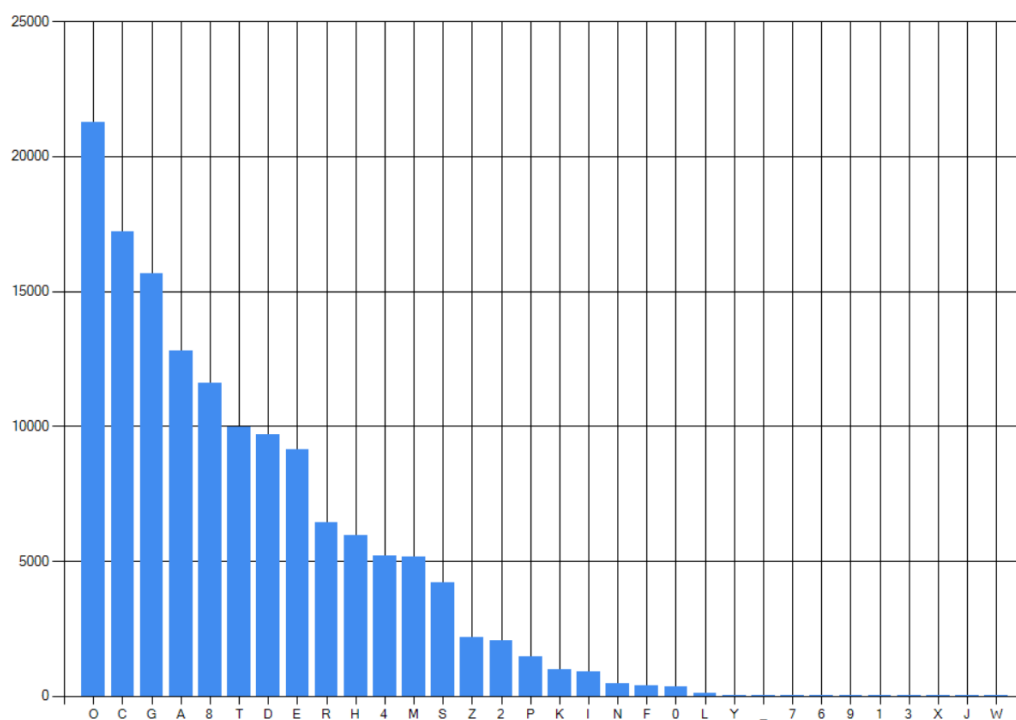
## 4 Závěr

Cílem této práce bylo navrhnout a implementovat nástroj schopný přečíst přepis rukopisu s různými nepřesnostmi a adekvátně tato data z textového souboru zpracovat. Následně provést základní typ analýzy a vizualizovat výsledky.

Řešení jsem si rozdělil na několik částí, čímž jsem si chtěl práci rozdělit a zároveň seřadit od nejjednoduchých úkonů až po ty složitější jako bylo generování kombinací nejasných slov apod. To celé s tím, že GUI část celého projektu ponechám na konec. Základním pilířem bylo sestavit algoritmus, který přečte textový soubor a zpracuje data nahrubo, mimo nejasná slova (v této fázi je kompletně vynechával). Dopředu jsem si analyzoval problém ohledně řešení nejasností a vytýčil jsem si řešení tak, že místo přímo definovaných slov, jsem místa, kde má být slovo (nebo více slov), nazval buňkami. V další části jsem tedy řešil jak zpracovat nejasná slova a určil jakým způsobem s nimi nakládat v rámci celého textu přepisu. Tímto jsem již získal kompletní text přepisu. Výsledkem těchto kroků byla sestavena celá kniha, s kterou již bylo možné dále pracovat.

Tato část byla nakonec časově daleko náročnější, než jsem v začátcích celé práce očekával, ale myslím, že se mi problémy s tím spojené podařilo obstojně vyřešit.

Následně jsem sestavil jednoduché algoritmy prováděné nad získanými daty z přepisu, které provádí analýzu nad textem knihy. Zajímavé srovnání výskytu znaků (symbolů) v textu:



Obrázek 24: Statistická charakteristika výskytu znaků Voyničiny

Při porovnání například s češtinou můžeme vysledovat podobně se opakující křivku.

Na závěr jsem implementoval algoritmus, který vytvoří jakousi mapu průchodu celým prepisem. Slouží k hrubé představě poměru výskytů slov, jejich opakování apod. a následně v dalším výzkumu by měla sloužit jako pomůcka při návrhu řešení analýzy pomocí fraktálů, jejich predikcí atd.

Kdybych měl práci zhodnotit, tak i přes nečekané potíže se povedlo dovršit jistých cílů a například jsem si ověřil, že minimálně struktura použitých symbolů odpovídá dnes používaným jazykům, což poukazuje na použití substituční metody s jedinou abecedou, avšak tato metoda je velmi jednoduchá. Tím tedy žádná „teorie“ končit nemůže. Věřím, že Voynichův rukopis nás ještě hodně překvapí. Data je dále možné porovnávat nejen v rámci znaků, ale celých slov dokonce například s češtinou, jak z jedné teorie plyne, že by rukopis mohl být dokonce založen na českém jazyce.

Ke konci bych rád zmínil, že platforma .Net byla dobrá volba a věřím, že mi implementaci mnohdy podstatně zjednodušila oproti jiným jazykům, ačkoliv věřím, že bude jazyk C# v jistých výpočtech o něco pomalejší než například C++. A to mě vede k možnosti přechodu k tomuto jazyku při případném pokračování v rámci diplomové práce.

# Použitá literatura

1. **Kuchartz, Udo.** Qualitative Text Analysis: A Guide to Methods, Practice and Using Software, SAGE Publications Ltd, 2014
2. **Patton, Q, Michael.** Qualitative Research & Evaluation Methods: Integrating Theory and Practice 4<sup>th</sup> Edition, SAGE Publications Ltd, 2015
3. **Zelinka, Ivan.** Evoluční a výpočetní techniky – principy a aplikace, BEN, Praha, 2008
4. **Čandík, M; Včelař, F; Zelinka, I.** Fraktální geometrie – principy a aplikace, BEN, Praha, 2006
5. **Swenson, Christopher.** Modern Cryptanalysis: Techniques for Advanced Code Breaking, John Wiley & Sons, 2012
6. **McNichol, A a Nelson, A, Jeffrey.** Handwriting Analysis: Putting It to Work for You, McGraw Hill Professional, 1994
7. **Lenkova, J.** Voynichův rukopis aneb Nejzáhadnější kniha světa, CAD PRESS, 2012

## Elektronické zdroje

8. **Zmeškal, O; Bzatek, T; Nezadal, M; Buchníček, M.** Harmonic and Fractal Image Analyzer ~ HarFA [Online] [Citace: 18. 4 2016] <http://www.fch.vut.cz/lectures/imagesci/download/harfasp.htm>.
9. **Tomatis, Mariano.** Soyga: the book that kills. *Blog of Wonders*. 2012 [Online] [Citace 18. 4 2016] <http://www.marianotomatis.it/blog.php?post=blog/20120423&section=english>
10. **Girolami, Andrea.** Look Inside the Extremely Rare Codex Seraphinianus, The Weirdest Encyclopedia Ever. *www.wired.com*. 2013 [Online] [Citace 18. 4 2016] <http://www.wired.com/2013/10/codex-seraphinianus-interview/>
11. **Gardham, Julie.** Book of the Month: Hypnerotomachia Poliphili. *Glasgow University Library*. 2004 [Online] [Citace: 18. 4 2016] <http://special.lib.gla.ac.uk/exhibns/month/feb2004.html>
12. **Colavito, Jason.** Nazi Archaeology and the Oera Linda Book. *Jason Colavito*. 2014 [Online] [Citace: 18. 4 2016] <http://www.jasoncolavito.com/blog/nazi-archaeology-and-the-oera-linda-book>
13. **Lanxon, Nate.** Science Museum discovers Philosopher's Stone Scroll. *www.wired.co.uk*. 2012 [Online] [Citace: 18. 4 2016] <http://www.wired.co.uk/news/archive/2012-04/17/science-museum-discovers-rare-alchemy-scroll/viewgallery/276082>
14. **Lorraine.** The Smithfield Decretals. *Interesting Pretties*. 2012 [Online] [Citace: 18. 4 2016] <http://interestingpretties.blogspot.cz/2012/11/the-smithfield-decretals.html>
15. **Lucille, Martin.** Rohonc Codex: Hungarian Enigma. *Parrot Time*. 2014 [Online] [Citace: 18. 4 2016] <http://www.parrottime.com/index.php?i=9&a=95>






16. **Diamond, A, Stephen.** Reading The Red Book: How C.G. Jung Salvaged His Soul. *Psychology Today*. 2011 [Online] [Citace: 18. 4 2016] <https://www.psychologytoday.com/blog/evil-deeds/201102/reading-the-red-book-how-cg-jung-salvaged-his-soul>
17. **Gray, Emma.** Portents and Prophecies: Conrad Lycosthenes, 1557. *Australian Museum*. 2012 [Online] [Citace: 18. 4 2016] <http://australianmuseum.net.au/blogpost/museullaneous/portents>
18. **Zandbergen, René.** Transcription of the text. *The Voynich Manuscript*. 2016 [Online] [Citace: 18. 4 2016] <http://www.voynich.nu/transcr.html>
19. **Microsoft.** .NET Framework 4.5. *msdn.microsoft.com*. [Online] [Citace: 18. 4 2016] [https://msdn.microsoft.com/cs-cz/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/cs-cz/library/w0x726c2(v=vs.110).aspx).

# Seznam příloh

**Příloha A:** Tabulka použitých znaků ve Voynichově rukopisu [18] ..... - 43 -

Součástí BP je CD.

Adresářová struktura přiloženého CD:

-  Program
-  Soubory pro program
-  Zdrojový kód

**Příloha A:** Tabulka použitých znaků ve Voynichově rukopisu [18]

Znak	Bennet	FSG	Currier	A	Bennet	FSG	Currier
†	D	4	4	˘	I	I	I
◦	O	O	O	⋈	IL	IE	G
δ	S	8	8	⋈⋈	IIL	IIE	H
9	G	G	9	⋈⋈⋈	IIIL	IIIE	1
2	Z	2	2	⋈˘	IQ	IR	T
⋈	L	AAE	E	⋈⋈˘	IIQ	IIR	U
⋈˘	Q	R	R	⋈⋈⋈˘	IIIQ	IIIR	0
⋈˘˘	CT	T	S	˘˘	U	L	D
⋈˘˘˘	ET	S	Z	˘˘˘	N	N	N
⋈⋈	H	H	P	⋈˘˘	M	M	M
⋈⋈⋈	P	P	B	⋈˘˘˘	IM	IIIL	3
⋈⋈˘	K	D	F	⋈˘		K	J
⋈⋈˘˘	F	F	V	⋈˘˘		IK	K
⋈⋈⋈˘	CHT	HZ	Q	⋈˘˘˘		IIK	L
⋈⋈⋈˘˘	CPT	PZ	W	⋈˘˘˘˘		IIIK	5
⋈⋈⋈˘˘˘	CKT	DZ	X	⋈˘˘˘˘		(6)	6
⋈⋈⋈˘˘˘˘	CFT	FZ	Y	⋈˘˘˘˘˘		(7)	7
˘	A	A	A	˘˘	Y	Y	(n)
˘˘	C	C	C	˘˘˘	V	V	(v)