

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-5384-5958

**URČOVANIE GENETICKÝCH PREDISPOZÍCIÍ
POMOCOU REGULÁRNYCH VÝRAZOV
DIPLOMOVÁ PRÁCA**

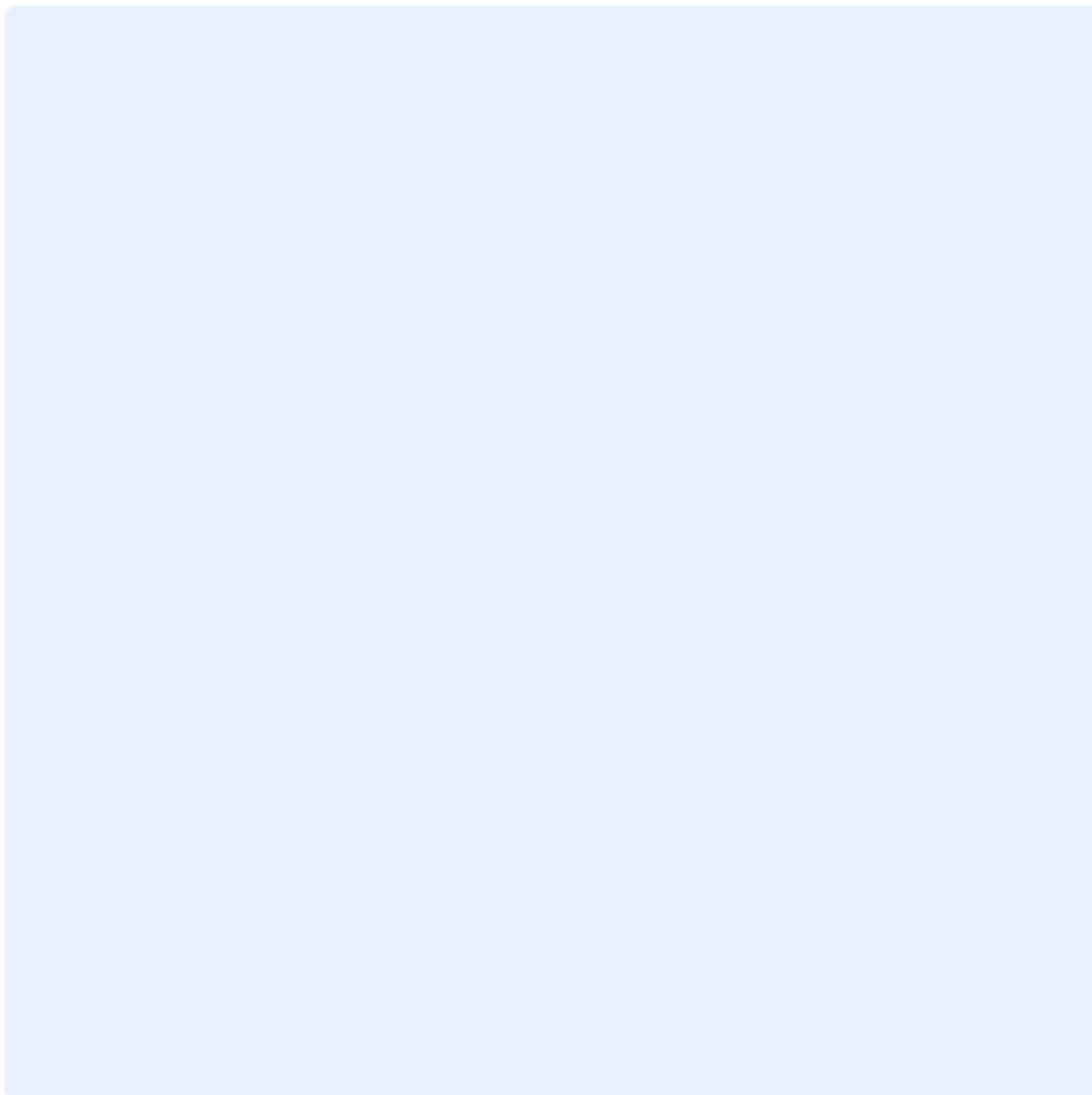
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-5384-5958

URČOVANIE GENETICKÝCH PREDISPOZÍCIÍ
POMOCOOU REGULÁRNYCH VÝRAZOV
DIPLOMOVÁ PRÁCA

Študijný program :	Aplikovaná informatika
Číslo študijného odboru:	2511
Názov študijného odboru:	9.2.9 Aplikovaná informatika
Školiace pracovisko:	Ústav informatiky a matematiky
Vedúci záverečnej práce:	Mgr. Zuzana Ševčíková

Sem vložte zadanie z AIS



SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program :	Aplikovaná informatika
Vyberte typ práce	Určovanie genetických predispozícií pomocou regulárnych výrazov
Autor:	Bc. Jakub Kanitra
Vedúci záverečnej práce:	Mgr. Zuzana Ševčíková
Miesto a rok predloženia práce:	Bratislava 2015

Vložte text súhrnu, ktorý obsahuje informáciu o cieľoch práce, jej stručnom obsahu a v závere abstraktu sa charakterizuje splnenie cieľa, výsledky a význam celej práce. Píše sa súvisle ako jeden odsek a jeho rozsah je spravidla 100 až 500 slov

Kľúčové slová: Sem vložte 3 - 5 kľúčových slov

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION
TECHNOLOGY

Study Programme:	Applied Informatics
Bachelor Thesis:	Vložte názov práce.
Autor:	Bc. Jakub Kanitra
Supervisor:	Mgr. Zuzana Ševčíková
Place and year of submission:	Bratislava 2015

Vložte text súhrnu, ktorý obsahuje informáciu o cieľoch práce, jej stručnom obsahu a v závere abstraktu sa charakterizuje splnenie cieľa, výsledky a význam celej práce. Píše sa súvisle ako jeden odsek a jeho rozsah je spravidla 100 až 500 slov

Key words: Sem vložte 3 - 5 kľúčových slov

Vyhlásenie autora

Podpísaný Bc. Jakub Kanitra čestne vyhlasujem, že som diplomovú prácu Určovanie genetických predispozícií pomocou regulárnych výrazov vypracoval na základe poznatkov získaných počas štúdia a informácií z dostupnej literatúry uvedenej v práci.

Uvedenú prácu som vypracoval pod vedením Mgr. Zuzany Ševčíkovej.

V Bratislave dňa 15.01.2015

.....

podpis autora

Pod'akovanie

Obsah

Úvod	1
1	Analýza problému 3
1.1	Genetika 3
1.1.1	Biológia bunky..... 4
1.1.2	DNA..... 5
1.1.3	Gén a mutácia 7
1.1.4	Projekty..... 7
1.2	Regulárne výrazy 8
1.2.1	Zápis 8
1.2.2	Nederministický konečný akceptor 10
1.2.3	Thompsonov konštrukčný algoritmus Error! Bookmark not defined.
1.2.4	Použitie 12
1.3	Distributívne systémy 13
2	Opis riešenia 16
3	Zhodnotenie 18
Záver	19
Zoznam použitej literatúry	20
Prílohy	I
Príloha A: Nadpis	II

Zoznam obrázkov a tabuliek

Figure 1 Obory potrebné pre riešenie diplomovej práce	3
Figure 2 Schéma eukaryotickej bunky. Zdroj: [3]	4
Figure 3 Karyotyp človeka. Zdroj: [3]	5
Figure 4 Štruktúra DNA makromolekuly	6
Figure 5 Nedeterministický akceptor	11
Figure 6 Deterministický akceptor	11
Figure 7 Nedeterministický akceptor s ϵ -prechodmi	11
Figure 8 Pravidlá Thompsonového konštrukčného algoritmu	13

No table of figures entries found.

In your document, select the words to include in the table of contents, and then on the Home tab, under Styles, click a heading style. Repeat for each heading that you want to include, and then insert the table of contents in your document. To manually create a table of contents, on the Document Elements tab, under Table of Contents, point to a style and then click the down arrow button. Click one of the styles under Manual Table of Contents, and then type the entries manually.

Zoznam skratiek a značiek

DNA – Deoxyribonukleová kyselina

RNA – Ribonukleová kyselina

HGP – Human Genome Project

API – Application Programming Interface

UI – User Interface

BRE – Basic Regular Expression

ERE – Extended Regular Expression

POSIX – Portable Operating System Interface

IEEE – Institute of Electrical and Electronics Engineers

RFC – Request For Comments

NKA – Nedeterministický konečný akceptor

DKA – Deterministický konečný akceptor

TKA – Thompsonov konštrukčný algoritmus

DS – Distributívny systém

DDoS – Distributed Denial of Service

Úvod

Genetickým poruchám a chorobám sa veľmi ťažko dá predísť a v posledných desaťročiach ich rapídne pribúda. Podľa štatistík 3-4% všetkých novorodencov trpí určitým genetickým defektom a spôsobuje 20% všetkých umrtí novorodencov. O vážnosti týchto ochorení značí aj fakt, že 10% všetkých dospelých a 30% detských hospitalizovaných pacientov má geneticky ovplyvnené choroby. [1]

Diagnostika týchto ochorení nieje jednoduchá, no vďaka rozsiahlemu štúdiu a analýze ľudskej DNA a následnom skúmaní génov vznikajú rozsiahle databázy, ktoré opisujú gény a ich mutácie, na ich základe sa časť z týchto chorôb dá diagnostikovať v počiatkoch a teda umožniť včasnú liečbu.

Rozmachu takejto diagnostiky ako jednej zo základných bráni finančná náročnosť a rýchlosť DNA sekvencizátora, teda prístroja ktorý zosekvencuje DNA na sekvenciu dusíkových báz ktoré opíšem v 1.1 . Tento prístroj je v relatívnych počiatkoch (prvý vytvoril Lloyd M. Smith v roku 1987) a už teraz vidíme rapídnu evolúciu, a teda sa môže očakávať nárast rýchlosti a dostupnosti tohto prístroja.

Ďalším dôležitým faktom je výpočtová náročnosť analýzy sekvencie, keďže kompletná ľudská DNA sekvencia má približne 3 miliardy nukleotidových párov a 25 000 – 30 000 génov ich určenie vyžaduje značnú výpočtovú silu. Práve na tento fakt sa táto práca zameriava.

Cieľom práce je vytvorenie systému, ktorý by pre danú DNA sekvenciu určil genotyp jedinca a teda určil aj prítomnosť dostupných chorôb. Pre vytvorenie takéhoto systému sú potrebné základné poznatky z molekulárnej biológie a genetiky, ale aj poznatky z teórie regulárnych výrazov a ich spracovávaní napríklad konečnými automatmi, tie sú opísané v 1.2. Táto práca sa neobmedzuje iba na DNA sekvencie, jej variácia sa môže použiť napríklad ako prostriedok big data analýzy napríklad na analyzovanie veľkých dát textov.

Rýchlosť a efektívnosť systému bude zabezpečovať jeho distribuovaná povaha. Tá zabezpečuje minimálne nároky na výpočtovú silu riadiaceho servera a spolieha sa na pripojené výpočtové zariadenia (uzly). Tie sú platformovo nezávislé, a teda to môžu byť tablety, smartfóny alebo počítače. Bližšie informácie o tomto type systémov je možné nájsť v 0. Tento druh bol vybraný pre jeho malé rozšírenie, no enormný potenciál, keďže správne navrhnutý môže byť efektívnejší ako superpočítač a takisto rozšírením dostupných

potenciálnych uzlov vďaka zrýchľovaniu zariadení a zväčšovaniu pokrytia vysokorýchlostným internetom.

Túto tému diplomovej práce som vymyslel nie z dôvodu, že som expert v nejakej zo spomínaných oblastí, no práve pre chuť a entuziazmus sa im venovať. Taktiež verím, že koncepty, algoritmy, výsledný systém a jeho podsystémy sa budú môcť uplatniť pri rôznych projektoch.

1 Analýza problému

Na Figure 1 sú v podobe diagramu ukázané potrebné znalosti na vyriešenie problému. V podkapitolách opíšem informácie a teóriu potrebné k splneniu zadania.

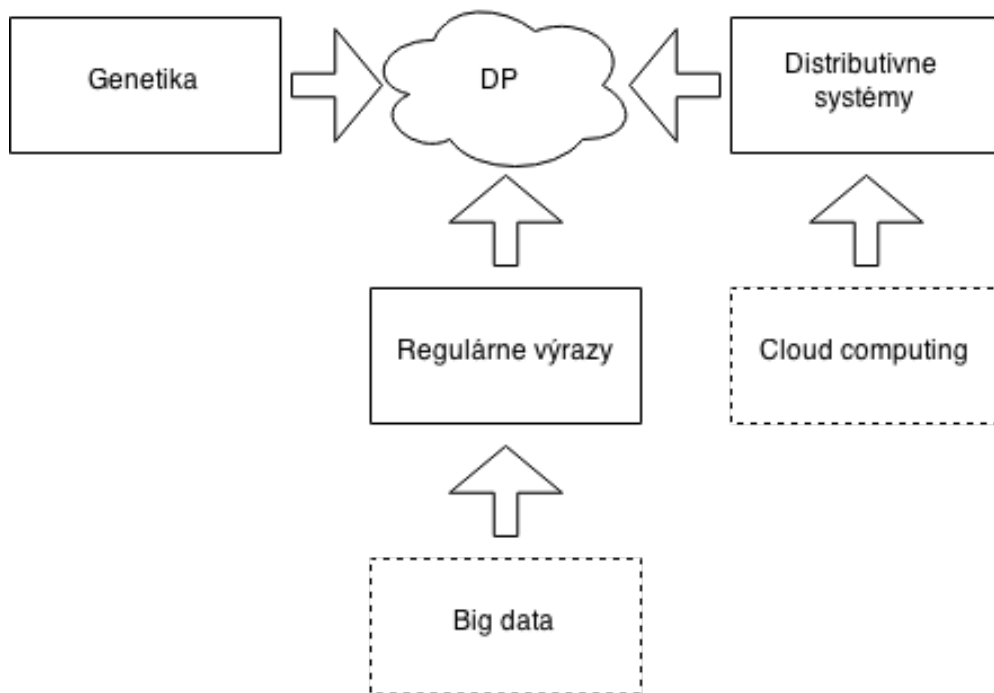


Figure 1 Obory potrebné pre riešenie diplomovej práce

1.1 Genetika

Táto časť je venovaná ozrejmieniu potrebných znalostí ohľadom genetiky a molekulárnej biológie, je nutné zdôrazniť, že práca je technického oboru a teda sa nebude opisovať do veľkej hĺbky, nebude tu spomenutá zložitá dedičnosť a replikácia jadra bunky a bunky samotnej. Tieto informácie sa dajú získať z knižných referencií [2] [3].

Genetika, veda o dedičnosti, je vo svojom základe štúdium biologickej informácie. Všetky živé organizmy, od jednobunkových baktérií, rastlín a zvierat, musia uchovať, replikovať a preniesť na potomkov mnoho informácií o vývoji, reprodukcii a prežití vo svojom prostredí. Genetici skúmajú ako organizmy odovzdávajú biologické informácie vo forme DNA na svojich potomkov a ako ich využívajú počas života. [2]

1.1.1 Biológia bunky

Aby sa mohla pochopiť súvislosť a neuveriteľná prepracovanosť živých tvorov, je nutné aby sa opísala základná stavebná jednotka organizmu, bunka.

Existujú 2 typy buniek určené podľa ich zloženia:

- Prokaryotické – neobsahujúce jadro, a preto sa DNA voľne pohybuje v cytoplazme bunky
- **Eukaryotické** (zobrazená na Figure 2) – obsahujúce jadro, DNA je pevne uložené vo vnútri a za žiadnych okolností ho neopúšťa, informácie sa prenášajú iba pomocou RNA, ľudské bunky sú tohto typu, a preto je tento typ v našom centre záujmu

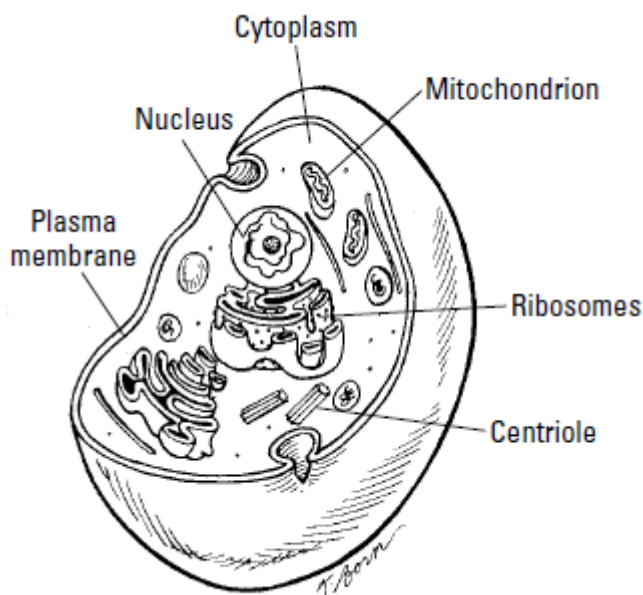
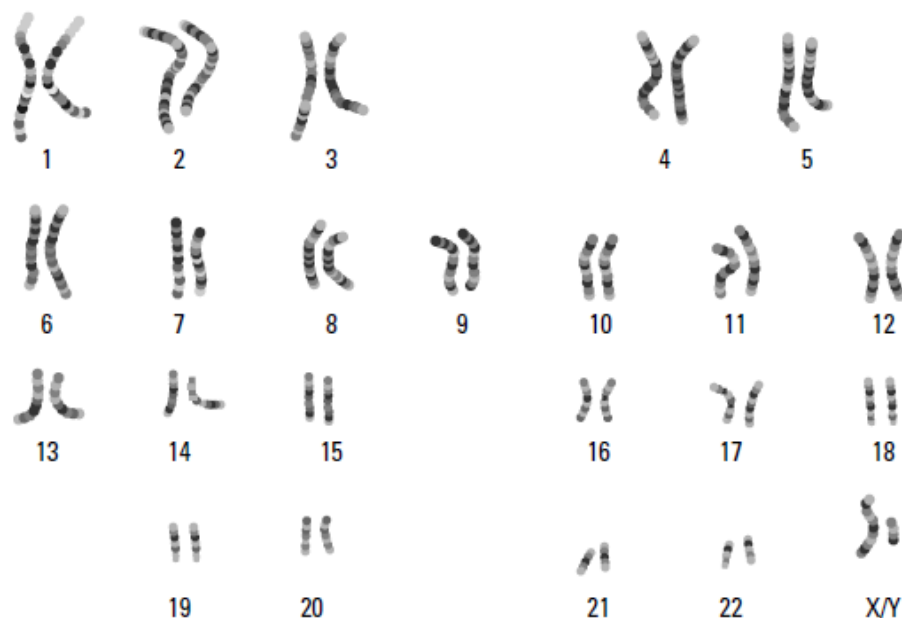


Figure 2 Schéma eukaryotickej bunky. Zdroj: [3]

Vo vnútri jadra sa nachádza genetická informácia v podobe chromozómov. Chromozóm je stužkovitý útvar pozostávajúci z DNA a pomocných naviazaných bielkovín. Nie je viditeľný ani pomocou mikroskopu, viditeľným sa stáva iba pri procese delenia bunky (mitózy). Počet chromozómov v jadre sa líši od druhu organizmu, človek má 46 chromozómov v jadre. Tie sa delia na 22 identických párov zhodujúcich sa v tvare a dĺžke a 1 pár pohlavných chromozómov, ktoré môžu byť zhodné (XX pre ženu) alebo rozdielne (XY pre muža). Ich unikátny tvar umožňuje presné zadefinovanie poradia chromozómov, čo je veľmi výhodné, keďže môžeme sekvenciu DNA zapísať ako

nepretržitý celok v definovanom poradí. Toto poradie sa nazýva aj karyotyp organizmu. Ukážka ľudského karyotypu je na Figure 3.



Normal Karyotype

Figure 3 Karyotyp človeka. Zdroj: [3]

Je nutné poznamenať, že nie každý organizmus obsahuje páry chromozómov, ale iba takzvané diploidné organizmy, napríklad osy sú haploidné čo znamená, že nemajú páry, ale sú organizmy, ktoré obsahujú až šesťnásť kópií toho istého chromozómu.

1.1.2 DNA

Ako bolo spomenuté, chromozómy sa skladajú z molekúl deoxyribonukleovej kyseliny (DNA). Tieto makromolekuly sú veľmi odolné a vďaka tomu je možné ich neporušenú extrakciu zo skamenelých kostí alebo zvierat umrznutých v ľadovcoch.

Chemické zloženie DNA molekúl je pomerne jednoduché. Dusíková báza, deoxyribózový cukor a fosfát sa spojí za vzniku nukleotidu. Nukleotidy sa označujú podľa použitej dusíkovej bázy. Zistilo sa, že v celej DNA sa nachádzajú iba 4 druhy dusíkových báz, konkrétne sú to adenín, guanín, tymín a cytozín. Tie sa komplementárne dopĺňajú, a vznikajú nukleotidové páry adenínu s tymínom a cytozínu s guanínom. Tisíce takýchto

párov sa skladajú do dvojitej závitnice zobrazenej na Figure 4. Táto závitnica je kľúčom k ochrane a veľkej odolnosti celej molekuly.

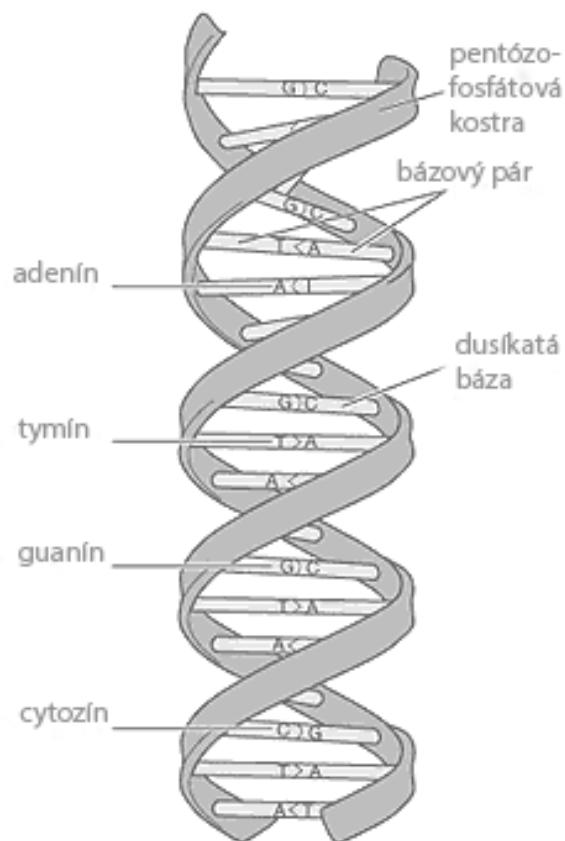


Figure 4 Štruktúra DNA makromolekuly

DNA molekuly zo všetkých ľudských chromozómov obsahujú približne 3 miliardy nukleotidových párov a kebyže sa tieto závitnice rozložia do radu, bol by dlhý zhruba 185cm [3]. Pre porovnanie baktérie majú zhruba 4 milióny nukleotidových párov.

Zápis celej alebo časti ľudskej DNA ako poradie nukleotidov sa nazýva **sekvencovanie**. Keďže nukleotidový pár je komplementárny stačí zápis iba jednej strany závitnice. Sú vyvinuté automatizované prístroje, ktoré používajú viacero techník na sekvencizáciu DNA, najznámejšia a najpoužívannejšia je Sangerova metóda.

Je nutné poznamenať, že zosekvencovanie celej ľudskej DNA, takzvané sekvencovanie genómu je časovo a finančne náročné. Avšak trend rapídneho klesania ceny a tým súvisiaceho času je vidno napríklad pri porovnaní ceny zosekvencovania celého genómu jednotlivca. V roku 2001 bola cena 100 miliónov dolárov no v roku 2014 to bolo menej ako 10 tisíc dolárov pri zachovaní menej ako 1% miery chybovosti, ktorá je komunitou akceptovaná [4].

1.1.3 Gén a mutácia

Gén je základná fyzická a funkčná jednotka dedičnosti. Gény môžu byť takzvané non-coding RNA a gény ktoré kódujú syntézu určitého druhu bielkoviny. Bielkoviny sú základné stavebné jednotky organizmu a zabezpečujú všetky fyzické a funkčné vlastnosti jedinca, napríklad farbu vlasov, očí, výšku no aj základné fyziologické vlastnosti ako trávenie a dýchanie. Všetky vlastnosti zakódované v DNA sa označujú ako fenotyp jedinca.

Vieme, že ľudský genotyp sa skladá z približne 21 000 génov. Tie sa líšia v dĺžke, a dosahujú až veľkosť 2,3 milióna nukleotidových párov. Niektoré vlastnosti sú ovplyvnené jediným génom (monogenické) a ovplyvnené skupinou génov (polygenické). Vedci odhadujú, že je 10 000 monogenických genetických ochorení, ako príklady uvedieme cystickú fibrózu alebo Huntingtonovu chorobu.

??Kódovanie bielkovín pomocou RNA – ak nebude dost' teórie??

Mutácia je permanentná zmena nukleotidovej sekvencie v DNA. Väčšina mutácií prebehne bez postrehnutia, pretože prebehne v takzvanom „junk DNA“, teda časti sekvencie, ktoré nepatrí do žiadneho génu. No ak prebehne v génovej sekvencii, dôsledky môžu byť fatálne. Napríklad na treťom chromozóme, ktorý nesie 1000-2000 génov sa nachádza jeden ktorý zabezpečuje syntézu rhodopsínu, svetlocitlivej bielkoviny nachádzajúcej sa na sietnici. Je zaznamenaných až 30 mutácií tohto génu, ktoré ovplyvňujú korektné videnie.

Typy mutácií na sekvenčnej úrovni sú [2]:

- Substitučné, nukleotidový pár sa posunie v sekvencii alebo sa obrátia jeho strany
- Odstránenie, odstránenie jedného alebo viacerých nukleotidových párov
- Vloženie, vloženie jedného alebo viacerých nukleotidových párov
- Inverzia, obrátenie poradia časti sekvencie

1.1.4 Projekty

Human Genome Project

Najväčšiemu skoku vo výskume genetiky vďačíme práve HGP [5]. Začal v roku 1990 a bol označený za úspešne ukončený v roku 2003. Jeho cieľom bolo zosekvencovanie celého ľudského genómu a určenie všetkých génov. Pri začiatkoch sa predpokladalo, že

existuje cca. 100 000 génov kódujúcich bielkoviny, no HGP potvrdilo, že ich je cca. 21 000 a zdokumentované ich uložilo vo verejných databázach.

Práve tento projekt odštartoval takzvanú genomickú revolúciu [5]. Vytvoril 310 000 pracovných pozícií a považuje sa za jeden z najväčších vedeckých prínosov v histórii ľudstva. Ovplynvil mnoho technologických a vedeckých odvetví od zdravotníctva, biotechnológií, poľnohospodárstva, veterinárstva, forenzných vied a mnohých iných.

Genome Browsers

V bioinformatike je nutný rýchly a spoľahlivý prístup k biologickým databázam za účelom získania genomických dát. Za týmto účelom bolo vytvorených viacero „genome browsers“, ktoré poskytujú API alebo UI pre získanie týchto dát. Väčšina obsahuje dáta z tých istých zdrojov a líšia sa iba vo forme.

Práve tieto dáta sú potrebné pre riešenie zadaného problému tejto Diplomovej práce a budú podrobnejšie opísané v Kapitole 0.

1.2 Regulárne výrazy

V Kapitole 1.1.2 bolo ukázané, že ľudská DNA a teda všetky vlastnosti jedinca sú zakódované do postupnosti 4 druhov nukleotidov, konkrétne sú to adenín (A), cytozín (C), tymín (T) a guanín (G). Preto sa ľudská DNA dá zapísať ako postupnosť týchto 4 znakov o veľkosti tri miliardy. V kapitole 1.1.3 zas boli opísané gény a mutácie vo vzťahu práve k tejto postupnosti. Z týchto poznatkov môžeme vyvodiť, že určenie génu je vlastne zistenie prítomnosti daného vzoru v reťazci. Tento vzor môže zakomponovávať rôzne variácie reťazca, v tejto implementácii sú to mutácie. Práve na takýto účel boli vytvorené regulárne výrazy.

„Regulárny výraz je zápis popisujúci množinu znakových reťazcov. Keď konkrétny reťazec je v množine popísanej regulárnym výrazom, tak sa hovorí, že reťazec vyhovuje vzoru.“ [6] Je široko používaný v teoretickej počítačovej vede a teórií formálnych výrazov.

1.2.1 Zápis

V najjednoduchšej forme môže regulárny výraz definovať konkrétne slovo alebo znak, no môže definovať aj zložité vzory napríklad validnú e-mailovú adresu, telefónne číslo, dátum alebo aj cystickú fibrózu v ľudskej DNA.

Časť IEEE POSIX štandardu, BRE (basic regular expression) a ERE (extended regular expression) zjednocuje zápis regulárnych výrazov. Využíva konvenčnú znakovú sadu a definuje metaznaky s kontrolnými funkciami.

Takýmito metaznakmi sú:

- `.` – ľubovoľný znak
- `[]` - označuje ľubovoľný znak z množiny vo vnútri, napríklad výraz `[abc]` značí prítomnosť znaku `a` alebo `b` alebo `c`, môže sa použiť aj skrátený zápis `[a-c]`
- `[^]` – negácia množiny vo vnútri zátvoriek
- `^` - začiatok textu alebo riadku
- `$` - koniec textu alebo riadku
- `()` – definovanie podvýrazu
- `\n` – vloženie `n`-tého podvýrazu definovaného spôsobom popísaným vyššie
- `*` - označuje výskyt predchádzajúceho elementu nula alebo viackrát, môže sa použiť s podvýrazom, napríklad `(abc)*` vyhovuje `""`, `"abc"`, `"abcabc"` atď.
- `{m, n}` – označuje výskyt predchádzajúceho elementu minimálne `m`-krát vrátane a maximálne `n`-krát vrátane, napríklad `a{2,3}` vyhovuje slovám `"aa"` alebo `"aaa"`
- `?` – označuje výskyt predošlého elementu nula alebo jedenkrát
- `+` – označuje výskyt predošlého elementu jeden alebo viackrát
- `|` – logické alebo

V praxi sa používajú aj takzvané **znakové triedy**. Existuje viacero znakových tried, napríklad malé písmená, všetky alfanumerické znaky, číselné znaky a ďalšie. Zápis týchto tried nieje ustálený a preto tu spomeniem iba najpoužívanejšiu implementáciu pomocou ASCII znakov najdôležitejších tried. Sú to:

- `[a-z]` – malé písmená
- `[A-Z]` – veľké písmená
- `[0-9]` – číselné znaky
- `[A-Za-z0-9]` – všetky alfanumerické znaky
- `[A-Fa-f0-9]` – znaky hexadecimálneho čísla

Je nutné poznamenať, že regulárne výrazy niesú obmedzené iba na ASCII znaky a môžu sa použiť aj znakové triedy unicode znakov použitím /u a unicode kódu znaku. Napríklad [/u00C1-/u01C4] obsahuje všetky špeciálne slovenské písmená.

Príklad

Demonštrácia zostrojenia regulárneho výrazu je možná na príklade overenia korektnosti zadanej e-mailovej adresy.

Štandardizovaná syntax emailovej adresy je definovaná v RFC 5322. Keďže emailová adresa je case-insensitive, je možné celý vstup pretransformovať na malé písmená. Syntax podľa RFC 5322 sa môže definovať ako *local-part@domain*, kde doménová časť musí obsahovať minimálne dve doménové úrovne a musí sa ukončiť top-level doménom, ktorá sa skladá z dvoch až šiestich ascii znakov.

Najintuitívnejší výraz, zabezpečujúci validitu e-mailovej adresy by mohol byť:
$$^{[a-z0-9.]+}@^{[a-z0-9.]+}\.^{[a-z]{2,6}}\$$$

Tento výraz má zopár nedostatkov, prvým je nepodporovanie všetkých povolených znakov aj keď všetky sa v praxi nepoužívajú. Dokonca emailová služba spoločnosti google nepodporuje všetky znaky a lokálna časť sa môže skladať iba z [a-z0-9-_'].

Takisto adresa sa nemôže začínať ani končiť bodkov a nemôžu sa v nej nachádzať za sebou stojace bodky.

Kompletný regulárny výraz napísaný v javascriptovej syntaxe [7]:
$$^{[w!#\$\%&'*/+=?`{|}~^-]+(?:\.[w!#\$\%&'*/+=?`{|}~^-]+)*@(?:[A-Z0-9-]+\.)+[A-Z]{2,6}}\$$$

1.2.2 Konečný akceptor

Pre nájdenie najefektívnejšieho spôsobu vytvorenia a vyhodnotenia regulárnych výrazov je nutné oboznámiť sa so základmi teórie automatov. Táto teória je jedna zo základných častí teoretickej počítačovej vedy a diskkrétnej matematiky.

??Spomenúť regulárne jazyky a gramatiky??

Automat je abstraktný model stroja, ktorý vykonáva spracovanie vstupu pomocou pohybu po stavoch alebo konfiguráciách. V každom stave prechodová funkcia určí nasledujúci stav alebo konfiguráciu z konečnej množiny stavov a konfigurácií. [8]

Najkomplexnejšou aplikáciou teórie automatov je Turingov stroj, hypotetický automat manipulujúci so znakovou páskou a množinou pravidiel. Logika ľubovoľného počítačového algoritmu sa dá popísať Turingovým strojom.

Z rozsiahlej oblasti teórie automatov sa oboznámime s konečnými akceptormi. Konečný akceptor sa dá znázorniť ako orientovaný graf, nazývaný stavový diagram, kde miesta znázorňujú stavy a orientované hrany ohodnotené vstupným znakom abecedy definujú prechodovú funkciu δ . Miesto do ktorého vedie prázdna šípka je počiatočný stav a miesta označené dvojítm krúžkom definujú množinu koncových stavov. Príklady stavových diagramov konečných akceptorov akceptujúcich regulárny výraz $a(bb)^+a$ sú na Figure 5 , Figure 6 a Figure 7.

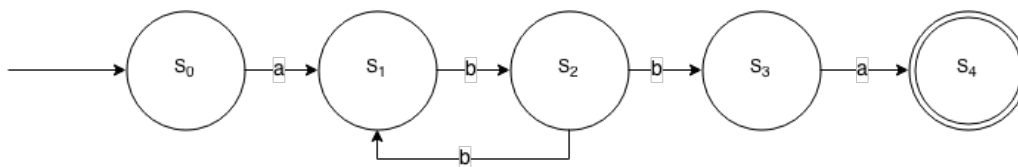


Figure 5 Nedeterministický akceptor

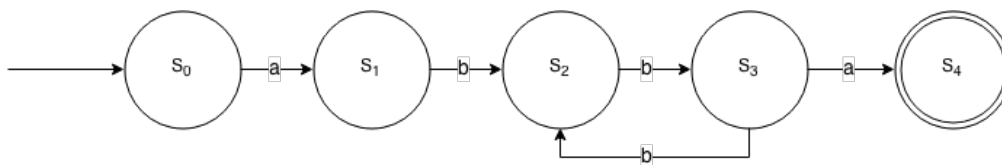


Figure 6 Deterministický akceptor

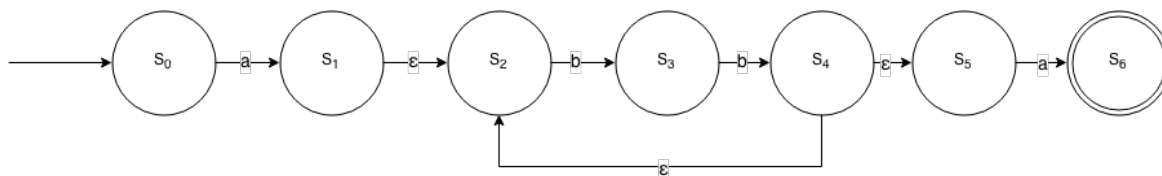


Figure 7 Nedeterministický akceptor s ϵ -prechodmi

Keď nový stav je určený súčasným stavom a vstupom, potom je akceptor deterministický (DKA), zo stavového diagramu sa determinizmus dá určiť, že zo žiadneho stavu nevedú dve a viac šípok ohodnotených rovnakým vstupom. Nedeterministický akceptor (NKA) takéto obmedzenie nemá. Existuje forma zápisu nedeterministického automatu pomocou takzvaných ϵ -prechodov. ϵ -prechod dáva ešte väčšiu voľnosť pri zobrazovaní, pretože jeho použitie nieje podmienené vstupom.

Každý NKA sa dá vyjadriť pomocou DKA, ktorý rozpoznáva rovnaký formálny jazyk. DKA je značne rozsiahlejší (ak NKA má n stavov, DKA ekvivalent môže dosahovať až 2^n stavov). Na konverziu sa môže použiť *Rabin-Scott powerset construction* algoritmus.

Testovanie pomocou je DKA je intuitívnejšie, no výpočtové systémy vďaka schopnosti rekurzie dobre pracujú aj s NKA. Bližšie je tento proces popísaný v časti 1.2.3.

Nedeterministický konečný akceptor (NKA) je konečný automat bez výstupnej funkcie a môže byť popísaný päticou: $SM = (S, E, \delta, s_0, A)$, kde [9]:

- S je množina stavov
- E je množina vstupných znakov (vstupná abeceda)
- $\delta: S \times E \rightarrow 2^S$ je prechodová funkcia priradujúca stavu a vstupu množinu nasledujúcich stavov
- $s_0 \in S$ je počiatočný stav
- $A \subseteq S$ je množina koncových stavov

1.2.3 Thompsonov konštrukčný algoritmus

V predošlej časti bolo ukázané, že ľubovoľný regulárny výraz generujúci regulárny jazyk sa dá vyjadriť pomocou nedeterministického konečného akceptora. Existuje viacero algoritmov ako takýto akceptor zostrojiť. Jeden z najpoužívanějších je Thompsonov konštrukčný algoritmus, publikovaný Kenom Thompsonom v roku 1968.

Algoritmus je založený na rekurzívnom rozklade výrazu na jeho podvýrazy až na elementárne výrazy. Tie sa zapisujú pravidlami znázornenými na Figure 9. Kde e je blok pozostávajúci z ďalších blokov alebo z elementárnych výrazov. Výraz z 1.2.2 po vytvorení pomocou TKA je na Figure 8.

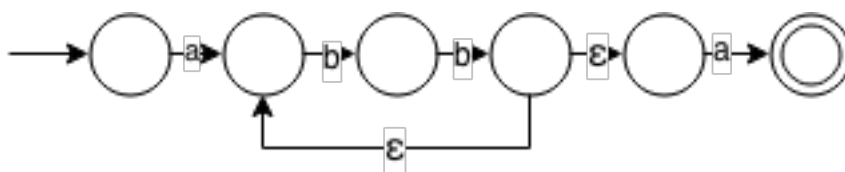


Figure 8 Diagram príkladu podľa Thompsonovho konštrukčného algoritmu

TKA vytvára ϵ -prechody, ktoré sú výhodné pre spracovanie na počítačoch, vďaka novej rekurzii. Vždy keď sa narazí na stav z ktorého vychádzajú ϵ -prechody dôjde k rekurzívnomu rozvetveniu.

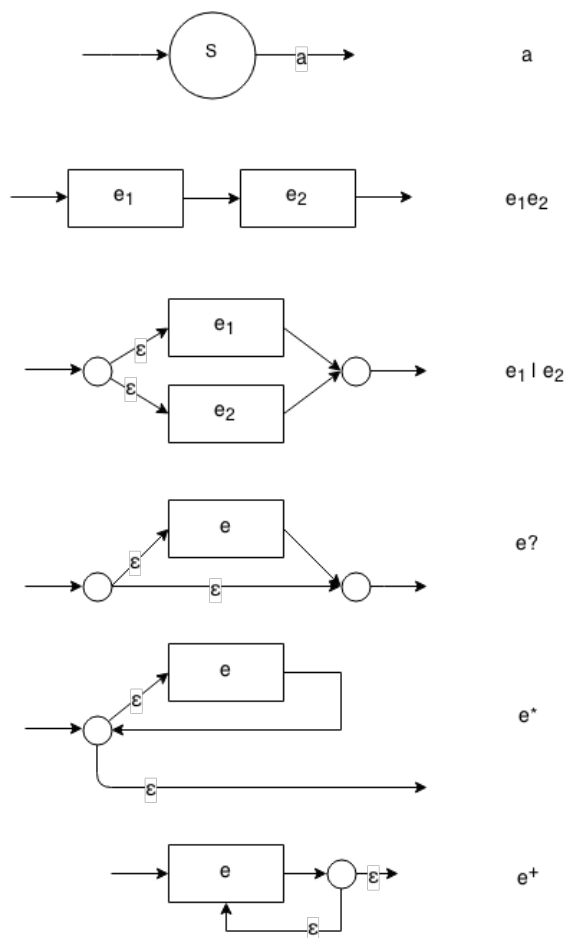


Figure 9 Pravidlá Thompsonového konštrukčného algoritmu

1.2.4 Použitie

S regulárnymi výrazmi sa človek stretáva každodenne, či je to vyhľadávanie textu na webovej stránke alebo v dokumente, alebo parsovanie html stránky prehľadávačom a mnohých iných.

Regulárne výrazy sa taktiež používajú ako jeden zo základných kameňov big data analýzy, ktorá sa v posledných rokoch stáva dominantným informačným artiklom. Či už je to určovanie trendov zo sociálnych sietí ako twitter alebo facebook, alebo ochrana pred kriminálnymi činnosťami analyzovaním komunikačných sietí bezpečnostnými úradmi. Taktiež väčšina takzvaných NO-SQL databázových systémov využíva prednosti regulárnych výrazov.

Na tomto ešte popracovať

1.3 Distribuované systémy

Jedným z výstupov tejto diplomovej práce je vytvorenie rozhrania distribuovaného systému (DS), ktoré je možné implementovať na rôzne výpočtové úlohy a problémy. Aby bolo toto rozhranie možné navrhnuť a implementovať je nutné oboznámiť sa s distribuovanými výpočtami ako oborom počítačovej vedy, ktorý sa zaoberá práve štúdiom DS.

1.3.1 Základy

„Distribuovaný systém je kolekcia nezávislých počítačov, ktoré sa javia používateľom systému ako jeden počítač.“ [10]

Motivácia k vytváraniu a používaniu distribuovaných systémov leží v túžbe zdieľania zdrojov. Zdroj je abstraktný pojem a charakterizuje všetky veci, ktoré môžu byť zdieľané v počítačovom systéme, môžu to byť hardwarové komponenty ako procesor, pamäť, grafická karta, harddisk alebo tlačiareň až po softvérové entity ako súbory, zložky, databázy a iné dátové objekty. Taktiež sa môžu zdieľať dáta z kamier, napríklad dopravných.

S mnohými variáciami týchto systémov sa človek stretáva denne, či už sú to virtuálne decentralizované meny ako bitcoin alebo lifecoin, peer-to-peer aplikácie, multiplayerové hry ale najfundamentálnejším príkladom je internet. Bohužiaľ, výpočtová sila týchto systémov poskytla mocný nástroj záškodníckym činnostiam v podobe DDoS.

DDoS (Distributed Denial of Service) je útok na server alebo cluster serverov pomocou zasielania požiadaviek (request-ov) všetkými uzlami distribuovaného systému. Sila takýchto útokov bola použitá v júli 2009 [11], kedy systém o veľkosti 166 000 počítačov znefunkčnilo viacero systémov, medzi ktorými boli stránky Pentagonu, Bieleho domu a ďalších.

V roku 2017 sa podľa agentúry Gartner [12] predá 2.9 miliárd kusov výpočtovej techniky v podobe PC, notebookov, tabletov a smartfónov. Z toho sa dá dedukovať, že každý obyvateľ technologicky vyspelej krajiny disponuje nejakým výpočtovým zariadením, ktoré po väčšinu dňa nevyužíva. Prečo by sa tieto zariadenia nemohli použiť ako výpočtové jednotky distribuovaných systémov a tým pádom znížili dopyt po stále nových a nových zariadeniach, čo má negatívny dopad na ekonomiku jedinca a taktiež negatívny dopad na životné prostredie?

1.3.2 Výzvy

Distribúované systémy sú tak rozsiahle a môžu sa nimi implementovať jednoduché úlohy ako posielanie správ ale aj najkomplexnejšie systémy ako internet alebo výpočet predpovede počasia.

Pri návrhu a implementácii ľubovoľného distribúovaného systému sa podľa Coulourisa [13] naskytuje sedem hlavných výziev, ktoré architekt systému musí brať do úvahy:

- Heterogenita – prístup k systému z rôznych zariadení, výrobcov a pripojení
- Otvorenosť – úroveň možnosti rozšírenia a reimplementácie
- Bezpečnosť – zdieľanie a preposielanie citlivých údajov musí byť chránené pred útokmi, taktiež systém musí byť zabezpečený voči útokom ako DoS
- Škálovateľnosť – teoretická neobmedzenosť možných používateľov systému
- Spracovanie chýb – je nutné brať do úvahy rôzne chyby, ktoré sa môžu vyskytnúť počas chodu, napríklad odpojenie uzlu od siete, chyba harddisku a mnoho ďalších
- Konkurencia – prístup 2 a viacerých používateľov k rovnakému zdroju nemôže mať za následok nekonzistenciu systému
- Prieľadnosť – používateľovi sa systém javí ako celok, netuší nič o jeho distribúovanej povahe

Návrhu konkrétneho distribúovaného systému sa budem venovať v Kapitole 0.

1.3.3 Aplikácie

Existuje viacero distribúovaných výpočtových systémov postavených na fakte, že ľudia poskytujú výpočtový výkon svojich počítačov za pocit vedomia, že sa podieľajú na spoločensko-vedecky prospešných projektoch. Verím, že popularitu takýchto systémov by zvýšil projekt, ktorý by finančne motivoval používateľov k zdieľaniu ich výkonu.

Najúspešnejší projekt je podrobnejšie rozpísaný nižšie, no za zmienku stoja aj **MilkyWay@home** generujúci presný trojrozmerný dynamický model galaxie Mliečnej cesty, ktorý má priemerný výpočtový výkon 573 TFLOPS a 27 000 aktívnych používateľov a **GIMPS** (Great Internet Mersenne Prime Search) hľadajúci Mersenove prvočísla, ktoré sú definované vzťahom $M_n = 2^n - 1$.

Einstein@home

Tento najúspešnejší dobrovoľnícky projekt, ktorý využíva distribuovaný výpočtový systém hľadá slabé astrofyzické signály z rotujúcich neutrónových hviezd (pulzarov) s použitím dát z LIGO gravitačno-vlnových detektorov, rádiového teleskopu Arecibo a satelitu na detekciu gama žiarenia Fermi. Projekt od svojho počiatku v roku 2005 detkoval 36 neutrónových hviezd a jeho cieľom je potvrdenie existencie gravitačných vĺn emitovaných neutrónovými hviezdami. Tieto vlny predpovedal Albert Einstein, no ešte neboli nikdy priamo detekované.

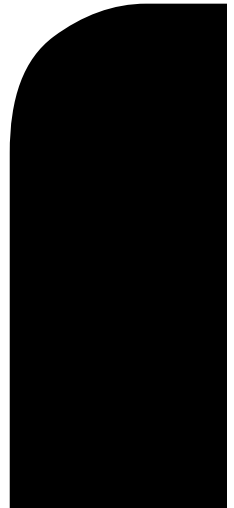
Priemerná výpočtová sila tohto distribuovaného systému je 470 TFLOPS [14], čím by sa mohol zaradiť medzi prvých 500 superpočítačov na svete.

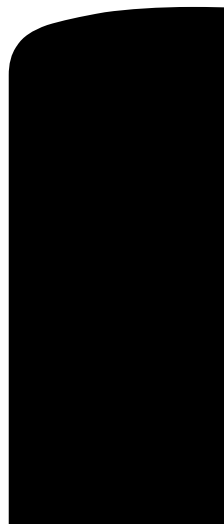
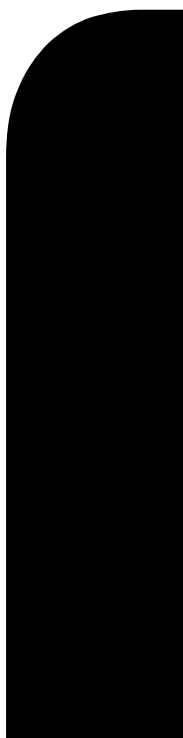
2 Opis riešenia

3 Zhodnotenie

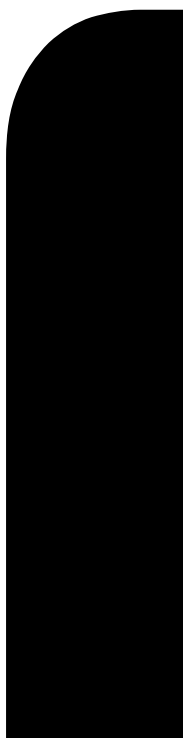
Záver

Zoznam použitej literatúry







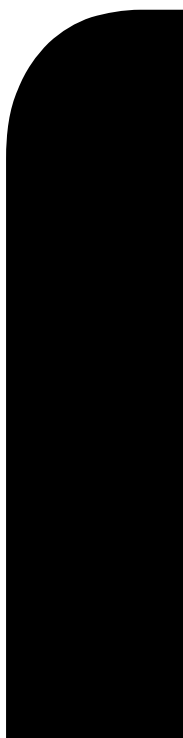


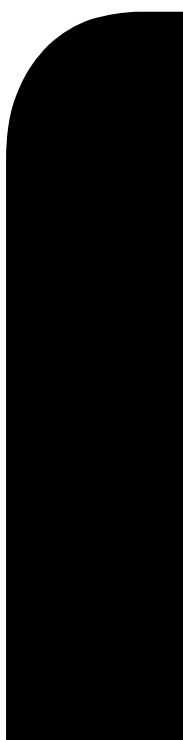


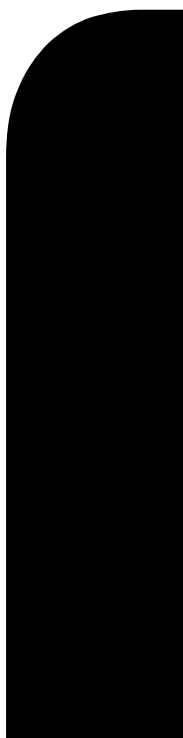
7



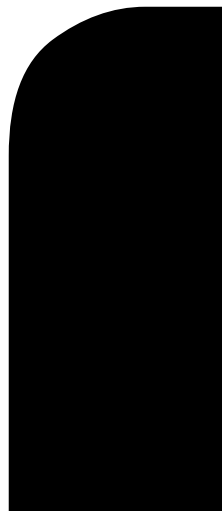








1

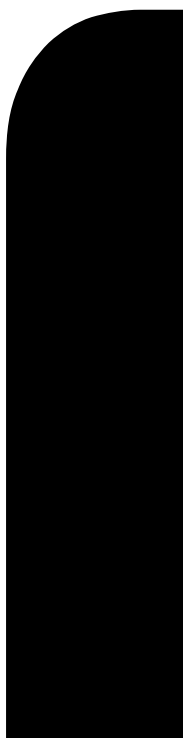






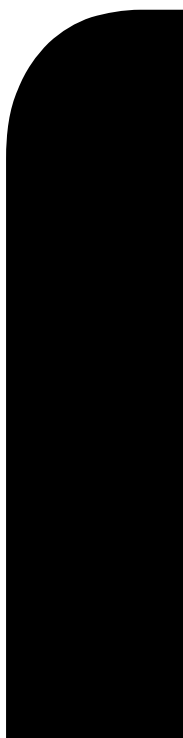
3





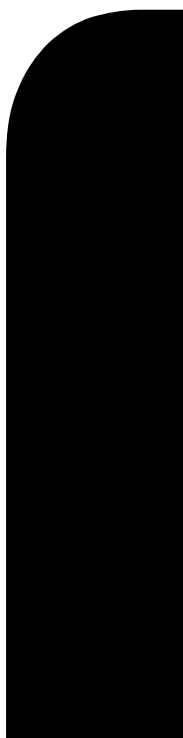


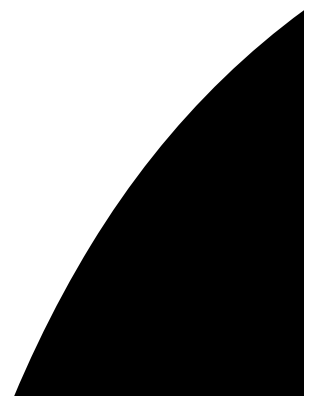




7









Prílohy

Príloha A: Nadpis.	II
-------------------------	----

Príloha A: Nadpis