VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

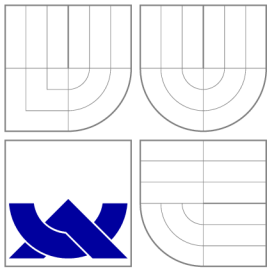
MORFOLOGICKÝ ANALYZÁTOR POMOCÍ KONEČNÝCH AUTOMATŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

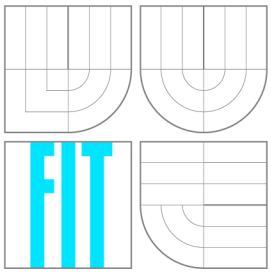
AUTOR PRÁCE
AUTHOR

STANISLAV ČERNÝ

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MORFOLOGICKÝ ANALYZÁTOR POMOCÍ KONEČNÝCH AUTOMATŮ

MORPHOLOGICAL ANALYSER IMPLEMENTED AS FSAS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

STANISLAV ČERNÝ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2008

Abstrakt

V textu je popsán morfologický analyzátor češtiny, který využívá slovníkový přístup. Slovník je uložen pomocí deterministického konečného automatu. Další část textu je zaměřena na analýzu číslovek, a to zejména na získávání numerických hodnot, které reprezentují. Vedle analýzy slov je nastíněna podpora pro generování vazeb mezi základními tvary.

Klíčová slova

morfologický analyzátor, TRIE, deterministický konečný automat, číslovky, slovotvorné vazby

Abstract

We describe morphemic analyser using dictionary approach. Dictionary is saved as deterministic finite state automata. Another part of text deals with analysis of numerals, especially retrieving numeric value from that words. Besides the analysis there is description of generating word-formation relationships in this work.

Keywords

morphemic analyser, TRIE, deterministic finite state automata, numerals, word-formation

Citace

Stanislav Černý: Morfologický analyzátor pomocí konečných automatů, bakalářská práce, Brno, FIT VUT v Brně, 2008

Morfologický analyzátor pomocí konečných automatů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením doc. Pavla Smrže. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Stanislav Černý
13. května 2008

Poděkování

Zde bych chtěl poděkovat svému vedoucímu doc. Pavlu Smržovi za bezpočet podnětů, rad a nápadů, kterými mě zásoboval během tvorby bakalářské práce. Dále mu chci poděkovat za poskytnutou literaturu.

© Stanislav Černý, 2008.
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Kapitola 1

Úvod

Tato práce se zabývá morfologickým analyzátorem češtiny. Hlavní funkcí analyzátoru je získání základního tvaru a mluvnické kategorie slova.

Ve vícejazyčných slovnících se slova uvádějí v základním tvaru, proto je při automatizovaném překladu mezi jazyky nutné hledané slovo nejprve převést na základní tvar, a až poté začít slovo vyhledávat v překladovém slovníku.

Mluvnické kategorie slova lze využít při implementaci „inteligentních“ korektorů češtiny, které budou do jisté míry schopny kontrolovat slovosled, či shodu podmětu s přísudkem.

Problém, který bylo nutno vyřešit, je velká prostorová náročnost. Vždyť čeština obsahuje více než 30 milionů tvarů slov, přičemž ke každému tvaru musí být přiřazen odpovídající základní tvar a různé morfologické kategorie. V oněch třiceti milionech slov jsou rozlišovány i tvary, které jsou sice reprezentovány stejnými hláskami, ale mají různý morfologický význam.

Kapitola 3 předvádí možnost, jak lze efektivně snížit počet tvarů, které musejí být uloženy ve slovníku, aniž by se snížilo pokrytí slov, které analyzátor poskytuje.

Kapitola 4 se zaobírá generováním slovotvorných vazeb. Tyto vazby spojují slova, která jsou odvozena ze stejného základu a mají tedy podobný význam. Vazby dále nesou sémantickou informaci o tom, jak se význam slova upravuje oproti významu základního slova.

Další problematikou, popsanou v kapitole 5, je analýza číslovek. Tento slovní druh má, již ze své podstaty, potenciálně mnohem větší počet tvarů, než zbylé slovní druhy dohromady. Tuto „explozi“ počtu tvarů mají na svědomí zejména číslovky, které jsou složeny z více kořenů. Text se snaží nalézt, jak tyto číslovky efektivně uložit, aniž by rychlost analýzy klesla pod únosnou mez.

Analyzátor u číslovek rozpoznává nejen jejich základní tvar a morfologickou kategorii, ale i číselnou hodnotu, kterou daná číslovka reprezentuje.

Teorie, zabývající se uložením slovníku, je stručně uvedena v kapitole 6.

Výslednou implementaci analyzátoru popisuje kapitola 7. Dále je zde představena knihovna libma, která celý analyzátor zapouzdřuje a svým rozhraním umožňuje snadný přístup ke všem jeho funkcím.

Jelikož autor nemá lingvistické vzdělání, je celý text doplněn množstvím názorných příkladů, které snad přispějí ke snadnějšímu pochopení problematiky a zabrání špatné interpretaci textu.

Kapitola 2

Morfologie

Morfologie, čili tvarosloví, je jedním z mnoha podoborů lingvistiky. Tato disciplína se zabývá stavbou slov, slovními druhy a postupy jak tyto druhy pravidelně odvozovat. Dále se zabývá významem těchto druhů [1].

Morfologie se dělí do dvou částí. Skloňováním a časováním ohebných slovních druhů se zabývá *flektivní morfologie*. Čeština patří mezi flektivní jazyky, tedy ty, které se ohýbají pomocí koncovek [8]. V některé literatuře je tvaroslovím myšlena právě flektivní morfologie.

Odvozováním slovních druhů se zabývá *derivační morfologie*. Slova se odvozují postupným přidáváním předpon a slovotvorných přípon. Toto odvětví se též nazývá slovotvorba.

2.1 Základní pojmy

Než přistoupíme k dalšímu výkladu, je nutné si sjednotit terminologii.

Kořen (root): Jedná se o slovotvorný základ slov. Kořen nese hlavní sémantickou informaci. V češtině se kořen při ohýbání slov mění jen ve výjimečných případech.

Předpona (prefix): Ve slově se nachází před kořenem. Předpona pozměňuje význam původního slova. Slovo může obsahovat i několik po sobě jdoucích předpon.

Infix: Nachází se za kořenem slova. Tato jednotka se v češtině vyskytuje jen zřídka. Slouží k zesílení významu přídavných jmen nebo příslovcí [8].

Interfix: Tato část se využívá při skládání slov. Jedná se o pomocnou jednotku, která nenese sémantickou informaci.

Přípona (suffix): Nachází se za kořenem. Podobně jako předpona mění význam slova, ale na rozdíl od předpony může slovo po přidání přípony změnit slovní druh. Každé slovo může obsahovat více přípon.

Výše popsaný typ přípon je v literatuře označován jako tzv. *kmenotvorné přípony*.

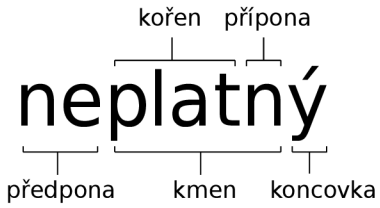
Koncovka (ending): Nachází se za příponami. Někdy je označována jako tzv. *tvarotvorná přípona*. Koncovky nesou informaci o pádu, číslu, rodu, osobě, atd. Proto se při skloňování a časování slov mění jen tato část slova.

Kmen (stem): Odstraněním koncovky ze slova získáme kmen slova. Kmen zahrnuje všechny kořeny spolu se slovotvornými příponami, které slovo obsahuje.

Příklad 2.1: Vztah mezi základními pojmy

kořen: ne–**let**–ět, **let**–adlo
předpona: **ne**–umět, **pře**–lézt
přípona: plav–**at**, plav–**b**–a, plav–**íc**–í se
infix: mal–**il**–inký
interfix: velk–**o**–město, tři–**a**–dvacet
koncovka: mlad–**ý**, mlad–**á**, mlad–**ého**

Diagram 2.1: Vztah mezi základními pojmy



Základní tvar (lemma): Jedná se o tvar slova, ve kterém je uváděn ve slovnících. Podstatná jména bývají uváděna v prvním pádu čísla jednotného. Přídavná jména se zapisují ve stejném pádu a čísle jako podstatná jména, základním rodem bývá rod mužský, používá se 1. stupeň. U sloves se uvádí tvar v infinitivu. Další ohebné druhy se zapisují obdobně. Neohebné slovní druhy mají jen jeden tvar, v němž jsou také uloženy ve slovnících.

2.2 Tvoření slov

Vlivem stálého vývoje vědy a technologie, vzniká potřeba nových pojmů a označení.

Jednou z možností je použití již existujících výrazů, kterým se přiřadí nový význam. Tento způsob lze používat jen omezeně, aby bylo vždy jasné, jaký z významů měl mluvčí na mysli.

Dalším způsobem rozšiřování slovní zásoby je přejímaní termínů z cizích jazyků. Některá z přejatých slov se v jazyku ujmou natolik, že jsou používána širokou veřejností. Ale většina z těchto výrazů je známá jen v úzkém kruhu odborné veřejnosti a pro zbytek populace je částečně nebo úplně nesrozumitelná.

Výše uvedené způsoby rozšiřování slovní zásoby byly uvedeny jen pro úplnost a dále se jimi nebudeme zabývat.

2.2.1 Odvozování slov

Slova se dělí na motivovaná a nemotivovaná. Motivovaná slova jsou ta, jejichž význam lze vysvětlit jiným slovem. Nemotivovaná slova takto vysvětlit nelze, tato slova slouží jen k označení věcí [5].

Příklad 2.2: Motivovaná a nemotivovaná slova

Motivovaná slova: *hřiště* je místo, kde se hraje
Nemotivovaná slova: *kámen*, *strom*

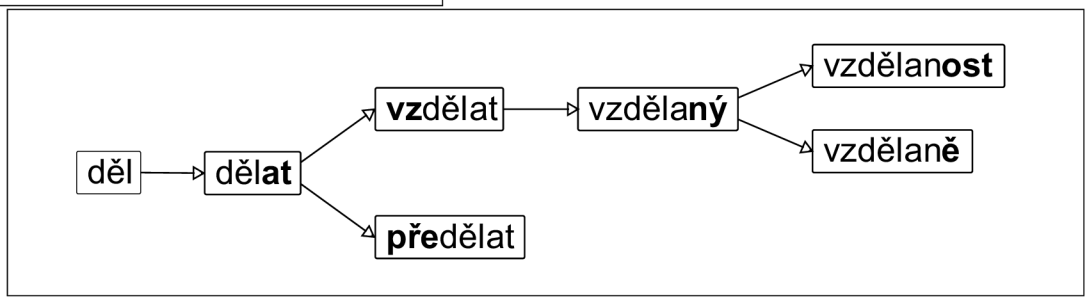
Motivovaná slova se odvozují pomocí prefixace a sufixace. Při prefixaci se ke slovu připojuje předpona, která mění význam slova. Sufixace je způsob, při kterém se ke kmeni připojuje slovotvorná přípona, která mění nejen význam slova, ale i mluvnickou kategorii slova. Přehled nejčastěji používaných předpon a přípon je uveden ve Slovníku spisovné češtiny [3].

Vztah mezi odvozenými slovy se nazývá fundace. Fundované slovo je to, které bylo odvozeno od slova výchozího. Výchozí slovo se označuje jako fundující.

Slova, která byla odvozena od společného základu patří do jednoho slovotvorného „hnízda“, přičemž odvozování mohlo proběhnout ve více než jednom kroku. Předpony a přípony vyjadřují, jaký vztah (fundace) je mezi výchozím a odvozeným slovem. Například přípona *ost* přidaná k přídavnému jménu vytvoří podstatné jméno, které bude popisovat vlastnost korespondující s významem původního slova.

Díky těmto vlastnostem odvozených slov stačí, k porozumění neznámých slov, znát jen význam některého slova z hnízda. Význam ostatních slov z hnízda lze odvodit díky zkušenosti, co jednotlivé předpony a přípony vyjadřují u slov v jiných hnízdech.

Diagram 2.2: Příklad části hnízda



Na diagramu 2.2 je znázorněna část slovotvorného hnízda spolu s vazbami mezi jednotlivými tvary. Fundace je znázorněna šipkou, která směřuje od fundujícího slova ke slovu fundovanému. Kořen je část slova, která je společná všem slovům v hnízdě. Přísně vzato, kořen do hnízda nepatří, protože se nejedná o regulérní slovo.

2.2.2 Skládání slov

Slova, která jsou vytvořena spojením více kořenů, se nazývají „složeniny“. Každý z kořenů přináší výslednému slovu svůj specifický význam. Jinými slovy lze říct, že význam výsledného slova je sjednocením významů jednotlivých kořenů. Díky tomu mají takto složená slova přesněji definovaný význam, čehož se využívá v odborných či geografických názvech. Tvoření slov tímto způsobem není v češtině tak časté jako v jiných jazycích (například v němčině).

Nejčastěji se vyskytují složeniny vytvořené z přídavných jmen. Při skládání těchto slov se vezme kmen z prvního slova, ke kterému se připojí slovo druhé. Mezi tyto části se často vkládá interfix, který zajišťuje snadnou výslovnost výsledného slova.

Příklad 2.3: Slova složená z více kořenů

Vodoměr: přístroj, který měří průtok vody.
Českomoravská vrchovina: geografický útvar na pomezí Čech a Moravy.

Příklad 2.4: Tvorba složeniny

velké město → velk–o–město

Definice 2.1 *První část složeniny může být chápána jako předpona [1].*

Díky této definici lze převést skládání slov na prefixaci, tedy způsob tvorby slov pomocí předpon. Tato skutečnost je využita v části 3.4, která popisuje tvorbu složených geografických názvů.

Kapitola 3

Předpony

V této kapitole budou představeny konkrétní předpony.

3.1 Negace

Zápor se vytvoří přidáním předpony *ne*. Tímto postupem lze vytvořit slovo s opačným významem z přídavných jmen, sloves, příslovcí a z některých jmen podstatných.

Pro vyjádření záporu lze také využít některé z antonym. Bohužel, ne všechna slova mají odpovídající antonymum. Navíc některá antonyma nejsou zcela přesným doplňkem původního výrazu. Za příklad vezměme slova *klesající* a *nerostoucí*. Jejich význam je velice podobný, ale každého matematika popudí, nebudeme-li rozlišovat mezi klesajícími a nerostoucími posloupnostmi.

Dalším důvodem pro upřednostnění tvaru s předponou může být stylistika. Někdy je použití antonyma společensky nevhodné. Porovnejme slova *ošklivá* a *nepěkná*. Každý cítí, že použití slova s předponou je společensky přijatelnější.

Při tvoření negací je nutné dávat pozor na některé výjimky. Tyto výjimky jsou shrnuty v příkladu 3.1. Podívejme se na problematiku z druhé strany, tedy na to jak z negovaného tvaru získat původní výraz. Musíme si uvědomit, že ne všechna slova začínající předponou *ne* jsou záporem nějakého kladného tvaru. Pro příklad takového slova uveďme výraz *nenávist*.

Příklad 3.1: Tvoření negací

pravidelné: **ne**dobrý, **ne**pracoval, **ne**hezky
výjimky: zápor slovesa *být* v 3. osobě: je → není
zkrácení samohlásky: brát → nebr**a**t

3.2 Superlativy

U některých přídavných jmen a příslovcí lze vyjadřovat míru kvality tzv. stupňováním. První stupeň (pozitiv) vyjadřuje vlastnost. Druhý (komparativ) slouží k porovnání předmětů. Třetí stupeň (superlativ) označuje nejvyšší míru.

Komparativ se nejčastěji tvoří pomocí přípon *ější*, *ší*. Superlativ vzniká sloučením předpony *nej* s druhým stupněm přídavného jména. Poznamenejme, že ne všechna slova

má smysl stupňovat, zejména slova vyjadřující látku, například *dřevěný*. Vedou se spory, zda lze stupňovat slovo *optimální*, které již samo vyjadřuje největší míru.

Některá přídavná jména se stupňují nepravidelně, ovšem tato nepravidelnost se týká jen tvoření komparativu. Naproti tomu je tvorba superlativů zcela pravidelná.

Příklad 3.2: Stupňování přídavných jmen

pravidelné: vysoký → vyšší → **nej**vyšší
nepravidelné: dobrý → lepší → **nej**lepší

3.3 Předpony přejaté z latiny

Zvláště v odborné češtině se používají předpony pocházející z latiny. Mezi ně patří předpony *hyper*, *super*, *supra* a *ultra*. Tyto předpony lze spojovat s podstatnými jmény, přídavnými jmény a příslovci.

Možná právě díky svému odbornému nádechu, se takto tvořená slova stále častěji vyskytují v médiích, odkud se dostávají do podvědomí širšího okruhu lidí.

Příklad 3.3: Předpony přejaté z latiny

supravodivost
ultrazvuk

3.4 Složené geografické názvy

Jedná se především o přídavná jména, která jsou složena z více kořenů, které označují geografické názvy. Všimněme si, že druhá část z těchto složenin jsou zároveň regulérní slova. Podle definice 2.1 lze označit část, která tomuto „vnořenému“ slovu předchází, jako předponu. Poznamenejme, že se tímto postupem za předpony označí i interfix. Takto získanou předponu lze připojit k jinému slovu, čímž získáme nový výraz.

Tento postup má hned několik slabin. Prvním problémem je získání vhodného seznamu prefixů. Druhá nepříjemnost se úzce dotýká sémantiky. Je zřejmé, že tyto předpony nelze skládat s libovolnými slovy. Výběr vhodných slov však závisí nejen na morfologické kategorii, ale i na významu slova.

Příklad 3.4: Složené geografické názvy

kanadskoamerická
českomoravská

3.5 Skládání předpon

Jak již bylo řečeno, slova se mohou skládat i z více než jedné předpony. Každá použitá předpona upravuje význam slova stejným způsobem, jako když je použita samostatně.

Stejně jako vše, má i skládání předpon svá pravidla. I když skladba české věty umožňuje použití více záporů, při stavbě slova toto neplatí. Třetí stupeň se tvoří spojením přípony *nej* a druhého stupně, proto je nemožné použít stejnou předponu dvakrát. Protože podruhé by byla předpona již aplikována na superlativ.

Dalším pravidlem je to, že prefix *nej* se vždy[1] nachází na nejlevější pozici ve slově. Obdobně je tomu při použití negace, jen s ohledem na předchozí pravidlo.

Pravděpodobně díky cizímu původu latinských předpon jsou pravidla pro jejich použití liberálnější, než u jiných předpon. Pomineme-li čistotu jazyka, je možné tyto prefixy umístit libovolně před i za výše uvedené předpony.

Příklad 3.5: Skládání předpon

chybné tvary: ne–ne–dobrý, nej–pěkný, ne–nej–pěknější
správné tvary: ne–dobrý, nej–pěknější, nej–ne–pěknější

[1] Výjimku tvoří předpony latinského původu, které jsou popsány v následujícím odstavci.

Kapitola 4

Slovotvorné vazby

Slovotvorné vazby byly částečně představeny v kapitole 2.2.1, která popisovala tvorbu slov odvozováním. Jednotlivé vazby reprezentují prefixy a sufixy, které mění význam fundovaného slova.

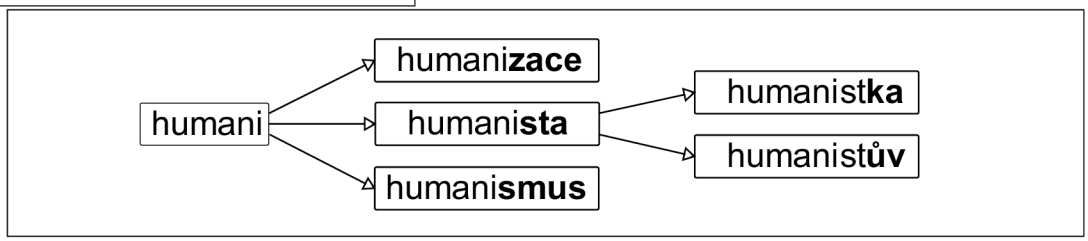
4.1 Slovotvorné vzory

Článek [5] popisuje, jak lze některé vazby seskupit tak, že vytvoří slovotvorný vzor – analogie k vzorům z klasické mluvnice, kterými se zabývá flektivní morfologie.

Klasické vzory obměňují slovo pomocí koncovek, což má za následek, že slovu po aplikaci vzoru zůstává stejný slovní druh i význam slova. Naproti tomu slovotvorné vzory mění slovo pomocí předpon a přípon, díky čemuž slovo mění nejen význam, ale i slovní druh.

Již zmíněný článek [5] uvádí jako příklad takového vzoru vzor *humanismus* Diagram 4.1, na kterém je tento vzor uvedený (oproti diagramu z původního textu je zjednodušen). Podle tohoto vzoru jsou vytvářena slova realismus, idealismus, apod.

Diagram 4.1: Vzor humanismus



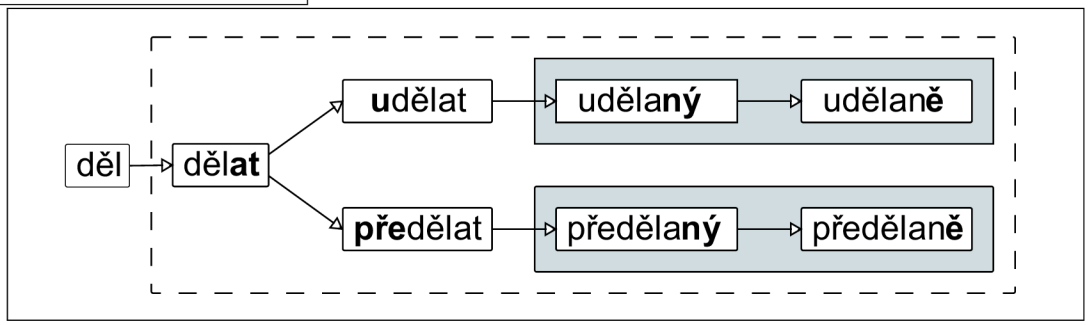
Na diagramu 4.2 je znázorněna část vzoru *dělat*. Slova, která lze odvodit od zmíněného vzoru jsou v čárkovaném obdélníku. V modrých oblastech jsou slova, která náleží vzoru *udělaný*. Je tedy zřejmé, že slovotvorné vzory lze dále skládat do hierarchie.

Pro získání všech slov, které lze odvodit z daného vzoru, je nutné postupně aplikovat všechny „podvzory“, které jsou v hlavním vzoru obsaženy.

4.2 Množiny vazeb

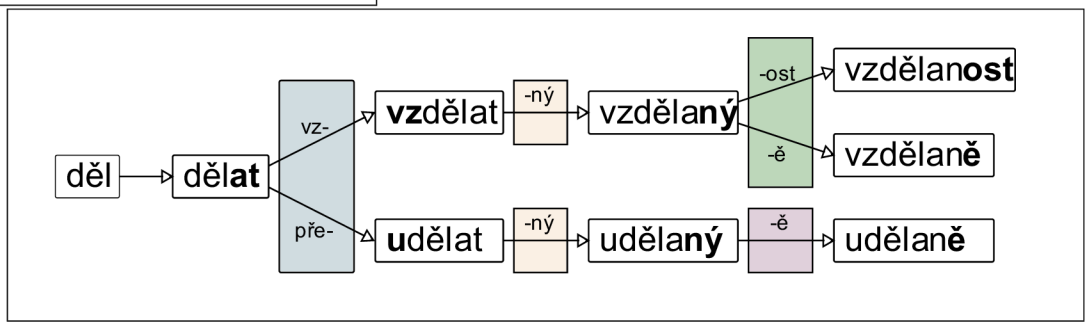
Z diagramů je patrné, že tyto vzory jsou tvořeny stromovitou hierarchií slovotvorných vazeb. Každá z těchto vazeb vytváří buď slovo, na které se již nebudou aplikovat další vazby nebo slovo, ze kterého lze odvozovat další slova.

Diagram 4.2: Podvzory



Všimněme si rozdílu mezi těmito vazbami a slovotvornými vzory. Jednotlivé vzory zahrnují všechny odvozené tvary od daného slova. Tedy vzor *humanismus* zahrnuje slova *humanismus*, *humanistický*, *humanizace*, atd. Na rozdíl od vzoru je množina vazeb skutečně jen seznam předpon a přípon, kterými jsou tvořeny další tvary. Pokud bychom množinu vazeb pojmenovali podle tvaru, na který ji lze aplikovat a jednotlivé vazby pojmenovali podle tvarů, které vytváří, pak by množina vazeb *humanismus* zahrnovala slova *humanistický*, *humanizace*, ale již ne tvar *humanismus*.

Diagram 4.3: Množiny vazeb



4.3 Analýza slova

Slovo je jednoznačně určené, pokud známe kmen, z kterého bylo odvozeno a posloupnost vazeb, které byly na kmen aplikovány.

Příklad 4.1: Analýza slova vzdělanost

Kmen: *děl*
Posloupnost vazeb: *-at* → *vz-* → *-ný* → *-ost*

Abychom zjistili, jaký vztah je mezi dvojicí slov, musíme udělat analýzu obou slov zvlášť, čímž získáme jejich kořeny a vazby, kterými byla vytvořena. Pokud jsou kořeny analyzovaných slov shodné, pak obě slova patří do shodného slovotvorného hnízda. Jestliže slova náleží do stejného hnízda, pak je jejich vztah přesně určen vazbami, kterými jsou tvořena. Čím více společných vazeb směrem od kořene dvojice slov má, tím více si jsou daná slova podobna významem.

Jako příklad vezměme dvojici slov *vzdělanost* a *vzděla*ně z diagramu 4.3. Obě slova patří do stejného hnízda, co víc, posloupnost vazeb, kterými jsou tvořena, se liší až v poslední vazbě, což značí, že daná slova mají velmi podobný význam. Podle rozdílných vazeb lze navíc vyvodit, co daná slova vyjadřují (*–nost* značí, že se jedná o vlastnost; *–ě* označuje příslovce).

Některá slova mohou být odvozena více způsoby. Ačkoliv je tvar výsledného slova shodný, význam těchto slov je jiný. Příklad 4.2 byl převzat z textu [8].

Příklad 4.2: Alternativní analýza slova

sval-ovec: parazit žijící uvnitř svalů
sval-ov-ec: expresivní výraz pro svalnatého muže

Kapitola 5

Číslovky

Tato kapitola bude popisovat slovní druh, který vyjadřuje počet či množství. Číslovky jsou jedním z nejproblematičtějších slovních druhů. I když se tento druh tradičně řadí mezi ohebné druhy, existují číslovky, které se neskloňují. Některé ohebné tvary se skloňují podle vlastních vzorů, jiné používají vzory z jiných slovních druhů, a to nejčastěji z přídavných nebo podstatných jmen. Dále existují číslovky, které mohou být zařazeny do jiných slovních druhů.

5.1 Druhy číslovek

Číslovky se dělí na určité a neurčité. Také je možno je rozdělit do druhů, podle toho, co označují. Mezi základními druhy jsou popsány tradiční číslovky. Po popisu základních druhů následují číslovky, které mohou být alternativně zařazeny do jiných slovních druhů.

5.1.1 Určité a neurčité

Určité číslovky popisují předmět kvantitativně, tj. nesou konkrétní hodnotu. Naproti tomu, číslovky neurčité se používají ke kvalitativnímu, vágnímu popisu. V některých situacích jsou i číslovky určité použity v přeneseném významu tak, že je lze považovat za číslovky neurčité. V přeneseném významu se jako číslovky používají i některá podstatná jména.

Příklad 5.1: Určité a neurčité číslovky

určité: dva; sto
neurčité: mnoho; málo
neurčité (číslovka určitá v přeneseném významu): říkat něco posté
neurčité (podst. jm. v přeneseném významu): moře slibů

5.1.2 Základní druhy číslovek

Základní (cardinalia): Tyto číslovky vyjadřují počet. Ptáme se na ně otázkou *kolik*.

- Mezi tyto číslovky patří:
  jedna; dva; deset; jedenáct; dvacet; jednadvacet; sto; tisíc; mnoho; několik; více
- Větší hodnoty se zapisují posloupností jednodušších číslovek oddělených mezerou:
  dvacet jedna; dvě stě; tisíc sto dvacet sedm

- Z pragmatických důvodů se částky na složenkách zapisují jedním slovem:
  ticíctřistaosmdesátšest

Řadové (ordinalia): Těmito číslovkami se vyjadřuje pořadí. Ptáme se na ně otázkou *kolikátý*. Tento druh je velice podobný přídavným jménům.

- Mezi tyto číslovky patří:
  první; prvý; desátý; jedenáctý; dvacátý; pětatřicátý; stý; tisící; miliontý; několikátý
- Jedním slovem se také zapisují:
  dvoustý; třináctistý; šestitísící; desetitisící; stotisící

Složitější číslovky se píší odděleně. Například číslo 7892. zapíšeme posloupností *sedmitisící osmistý devadesátý druhý* nebo *sedmitisící osmistý dvaadevadesátý*.

Násobné (multiplicativa): Tyto číslovky vyjadřují počet opakování nebo srovnání. Ptáme se na ně otázkou *kolikrát*. Jsou velice podobné přídavným jménům a příslovcím. Snadno se poznají podle toho, že to jsou složeniny, které v poslední části obsahují slova *krát* nebo *násobně*.

- Před částí krát se číslovky nacházejí v základním tvaru:
  dvakrát; desetkrát; stokrát; dvěstěkrát; tisíckrát; desettisíckrát; několikrát
- V některých textech se objevují i složeniny ve tvaru:
  stodvacetpětkrát

Prefix před *násobně* je ve stejném tvaru jako u složenin, které jsou probírány v jednom z následujících odstavců.

Druhové (specialia): Těmito číslovkami se vyjadřuje počet druhů. Ptáme se na ně otázkou *kolikero*. Tvoří se přidáním přípony *ero*.

- Příklad pravidelně tvořených číslovek:
  čtvero; desatero; devatenáctero; dvacatero; stero; tisícero; několikero
- Výjimkou jsou číslovky odvozené od hodnoty dva a tři:
  dvojí; trojí
- Složitější tvar složený do jednoho slova:
  sedmsetosmdesatero

5.1.3 Další číslovky

Zde popíšeme číslovky, které se někdy řadí do jiných slovních druhů, nejčastěji do podstatných nebo přídavných jmen.

Názvy číslic: Tyto číslovky slouží k pojmenování čísel. Ptáme se na ně otázkou *kolikátka*. Číslovky tohoto typu se snadno rozeznají podle poslední slabiky *ka*. Hojně se používají v hovorové češtině k vyjádření počtu nebo kvality.

Často zastupují podstatná jména v přeneseném slova smyslu. Například fotbalová *jedenáctka* označuje tým čítající jedenáct hráčů, *Dvanáctka* pro dvanácti stupňové pivo. *Jedničkou* je často myšlena tramvaj číslo jedna. Dalším tvarem může být *pětistovka*, která označuje bankovku s hodnotou pět set korun. Tato slova jsou také používána k rozlišení mezi typy produktů. Například Peugeot *tři sta šestka*.

Zlomky: Tyto číslovky označují hodnoty podílu. Opět je lze snadno rozlišit od jiných číslovek díky příponě *ina* na konci tvaru. Zlomky jsou tvořeny pravidelně až na tvary odvozené od číslovky *půl*.

- Mezi tyto číslovky patří:
  polovina; třetina; desetina; dvacetina; setina; tisícina; miliontina; dvoutřetinový

Složeniny: Tento typ slov má prefix tvořen pomocí kořenů reprezentujících číselnou hodnotu. Zbylá část je tvořena podstatným jménem, přídavným jménem nebo příslovcem. Tato část nese morfologickou informaci, proto tento typ slov není číslovkou, ale přejímá slovní druh od koncové části. Násobné číslovky obsahující kořen *násob* jsou tvořeny stejně jako zbylá slova z této skupiny.

- Mezi tato slova patří:
  trojúhelník; jedenadvaceticípý; tisícičlenný; stodvacetinásobně, pětiaktovka

5.2 Analýza číslovek

V následující kapitole bude popsán postup, jak analyzovat číslovky. Analýza číslovek vrací jejich morfologickou kategorii a případně číselné hodnoty, pokud analyzovaná slova patří mezi číslovky určité.

5.2.1 Tokenizace

Tokenizací je myšleno rozdělení složeného slova do jednodušších jednotek (tokenů, lexémů), kterými jsou v našem případě kořeny a interfixy. Nejdříve je nutné nalézt všechny tyto jednotky, nacházející se na začátku slova. Přípony a koncovku, které se jsou za posledním nalezeným kořenem, označíme jako *koncovou část*. Tato část nese informaci o druhu číslovky. Dále vyjadřuje morfologickou kategorii.

Ve formálních jazycích se při rozkládání textu na jednotlivé tokeny využívá specifických vlastností daného jazyka. Jako příklad uveďme název proměnné. V programovacích jazycích se tento lexém zapisuje většinou jako posloupnost znaků a číslic. Dostaneme-li se při čtení textu na posloupnost znaků, máme jistotu, že první nevyhovující znak, po této posloupnosti, označuje konec proměnné.

U přirozeného jazyka musíme zvolit jiný postup. Uložme si do seznamu všechny známé kořeny a interfixy. Analyzované slovo budeme procházet zleva. Pokud začátek slova odpovídá nějaké jednotce ze seznamu, uložme si tuto jednotku a pokračujme bezprostředně za touto částí. Jestliže v seznamu není odpovídající jednotka, bude zbytek slova koncovou částí.

Budou-li v seznamu tokeny *dva*, *dvanáct* a *dvacet*, je nutné zajistit, aby se při nalezení tokenu *dva* pokračovalo v hledání, zda se nejedná o token *dvanáct* či *dvacet*. Toto ovšem zbytečně komplikuje vyhledávání kořenů. Proto je pro uvedený algoritmus výhodné, aby v seznamu tokenů byly jen takové, které nejsou prefixem jiného.

5.2.2 Výpočet hodnoty

V následující části popíšeme postup výpočtu hodnoty, kterou reprezentuje analyzovaná číslovka.

Slova obsahující jednu číslici

Jestliže slovo obsahuje právě jeden kořen s číselnou hodnotou, pak celé slovo nabývá této hodnoty. Jedná se o nejjednodušší možnost.

Slova vyjadřující hodnotu od 10 do 99

Tuto problematiku rozložme na několik dílčích problémů.

Hodnoty od 11 do 19: Tyto číslovky se skládají ze dvou kořenů. První z kořenů označuje jednotky. Druhým je kořen *náct*, který vyjadřuje, že se k jednotkám má přičíst hodnota 10.

Desítky: I tyto číslovky jsou složeny ze dvou kořenů. Prvním je opět jednotka. Druhým kořenem *cet* je vyjádřeno, že má být jednotka vynásobena číslem 10.

Složené desítky: Jedná se o složeniny z jednotek a desítek. Tyto dvě části jsou spojeny pomocí interfixu *a*. Zde se nachází první problém. Hodnoty kořenů nelze zpracovávat v pořadí, v jakém se vyskytují ve slově. Nejprve se musí vyhodnotit desítky, které se poté přičtou k hodnotě prvního kořene.

Příklad 5.2: Výpočet hodnot u číslovek od 11 do 99

dva–náct	$\rightarrow$	$2 + 10$
dva–cet	$\rightarrow$	$2 * 10$
pět–a–dva–cet	$\rightarrow$	$5 + (2 * 10)$

Vyšší hodnoty

Kořeny *set*, *tisíc* a *milion* upravují řád předchozích číslovek. Pokud se číslovka skládá z více řádů, je nutné rozlišit řád, ke kterému jednotlivé číslice patří. Řády jsou skládány zleva od největšího po řád s jednotkami. Kořen označující řád je vždy posledním kořenem v řádu.

U sta tisíců lze pozorovat jakýsi vnořený řád stovek, který předchází tisícům.

Příklad 5.3: Výpočet hodnot u číslovek od 11 do 99

sto	$\rightarrow$	100
dvě–stě	$\rightarrow$	$2 * 100$
tři–náct–i–stý	$\rightarrow$	$(3 + 10) * 100$
sedm–set–dva–cet–tisíc–šest–set	$\rightarrow$	$[(7 * 100) + (2 * 10)] * 1000 + 6 * 100$

Zlomky

Většina zlomků je zakončena příponou *in*. Je-li ve slově obsažena tato koncovka, pak je nutné hodnotu číslovky převrátit. Kořen nepravidelné číslovky *půl* vyjadřuje hodnotu 0,5. Prefix zlomku může vyjadřovat hodnotu čitatele. Hodnotou čitatele vynásobíme výsledný zlomek.

Příklad 5.4: Výpočet hodnot u zlomků

pět–ina	$\rightarrow$	5^{-1}
dvou–třet–inový	$\rightarrow$	$2 * 3^{-1}$

5.2.3 Sémantické akce

Z předchozího textu vyplývá, že u kořene nestačí ukládat jen hodnotu, ale i operaci, která se má provést. Kvůli zlomkům je nutné ukládat informaci o operaci i u koncových částí.

Žádná operace: Tuto akci obsahují všechny interfixy a většina koncových částí. Označují se jí jednotky, které neobsahují informaci o hodnotě.

Hodnota: Tato akce je přiřazena ke kořenům jednotlivých číslic a značí, že se hodnota kořene má přičíst k mezivýsledku.

Řád: Tato akce se přiřazuje ke kořenům, které mění řád předcházejících číslic. Mezivýsledek se podle hodnoty řádu rozdělí na dvě části. Část s nižším řádem bude vynásobena hodnotou odpovídající řádu. Do mezivýsledku se uloží součet první části a nové hodnoty z druhé části.

Zlomek: Tuto akci obsahují koncové části, které označují zlomek. Do mezivýsledku se uloží jeho převrácená hodnota. Pokud byl ve zlomku i prefix označující čitatele, je převrácená hodnota vynásobena tímto čitatelem.

Neurčitá hodnota: Tato akce je nutná k označení kořenů s neurčitou hodnotou. Jestliže se v číslovce vyskytne alespoň jeden takovýto kořen, získává celá číslovka neurčitou hodnotu. Tato hodnota bude reprezentována hodnotou NAN[1] z reálných čísel.

5.2.4 Gramatika

Hlavní funkcí gramatiky je uložení korektních tvarů číslovek. Každý typ tvarů je reprezentován řádkem, který obsahuje posloupnost kořenů. Posledním členem této posloupnosti je označení koncové části, která může za daným tvarem následovat.

Kromě primární funkce lze gramatiku použít k úpravě priorit operací při výpočtu hodnot. První situací, kdy je potřeba upravit pořadí operací, je získávání hodnoty ze složených desítek. Zde je potřeba nejprve vypočítat hodnotu desítky a až poté k této hodnotě přičíst hodnotu první jednotky. Tohoto chování lze dosáhnout, pokud se první jednotka uloží do pomocné proměnné a do mezivýsledku se přičte až po zpracování řádu.

Další jev, který lze v gramatice lehce rozpoznat, je prefix zlomku obsahující čitatele. Opět lze prioritu snadno upravit tím, že se tento prefix uloží do pomocné proměnné. Po akci mající za úkol převrátit hodnotu mezivýsledku, se výsledná hodnota vynásobí hodnotou z pomocné proměnné.

[1] Not A Number

5.2.5 Algoritmus pro analýzu číslovek

Algoritmu se předává slovo, které je určené k analýze. Ze začátku tohoto slova se separuje kořen. V seznamu možných kořenů je uchována jeho sémantická akce a unikátní označení. Analyzované slovo se zkrátí o nalezený token, tím zajistíme, že se budeme dále zabývat jen dosud nezpracovanou částí slova. Poté se sémantickou akcí provede aktualizace hodnoty analyzované číslovky.

Jestliže je nutné pozměnit prioritu výpočtu, je tato skutečnost poznamenána bezprostředně za názvem kořene v gramatice. Pokud gramatika očekává, že další část slova může být koncovou částí, pak ověří, zda zbytek analyzovaného slova odpovídá některé z očekávaných přípon a koncovek. Seznam těchto částí je uložen obdobně jako seznam kořenů. Jestliže je nalezena odpovídající koncovka, pak se provede její sémantická akce a algoritmus končí.

Algoritmus 5.1: Analýza číslovek

1. Načti nový token.
2. **Když** byl načten nový token **a zároveň** tento token byl nalezen v gramatice.
3. Odstraň nalezený token ze začátku analyzovaného slova.
4. V gramatice se posuň na další kořen.
5. Proveď sémantickou akci.
6. **Když** je třeba změnit pořadí výpočtu hodnoty.
7. Ulož aktuální hodnotu do pomocné proměnné
8. **Když** je konec pravidla v gramatice
9. **Když** zbytek analyzovaného slova je jednou z koncových částí.
10. Proveď sémantickou akci koncové části.
11. Ukonči analýzu.
12. Pokračuj bodem 1.
13. **Jinak** ukonči analýzu.

5.2.6 Příklad získání hodnoty z číslovky

Pro ilustraci sémantických akcí uveďme příklad získání hodnoty hodnoty z čísla *sedmsetdvacettisícšestsetpětkrát* (720 605). Ve třetím sloupci tabulky je výraz, který se vypočítává po přijmutí lexému, jehož tvar je napsán v prvním sloupci.

Příklad 5.5: Získání hodnoty číslovky

kořen	**sémantická akce**	**úprava hodnoty**
sedm	hodnota 7	$0 + 7 \rightarrow 7$
set	řád 100	$(7 \mod 100) * 100 + \lfloor 7/100 \rfloor \rightarrow 700 + 0 \rightarrow 700$
dva	hodnota 2	$700 + 2 \rightarrow 702$
cet	řád 10	$(702 \mod 10) * 10 + \lfloor 702/10 \rfloor \rightarrow 20 + 700 \rightarrow 720$
tisíc	řád 1000	$(720 \mod 1000) * 1000 + \lfloor 720/1000 \rfloor$
		$\rightarrow 720\,000 + 0$
		$\rightarrow 720\,000$
šest	hodnota 6	$720\,000 + 6 \rightarrow 720\,006$
set	řád 100	$(720\,006 \mod 100) * 100 + \lfloor 720\,006/100 \rfloor$
		$\rightarrow 600 + 720\,000$
		$\rightarrow 720\,600$
pět	hodnota 5	$720\,600 + 5 \rightarrow 720\,605$

Kapitola 6

Datová struktura TRIE

V této kapitole popíšeme vyhledávací strukturu TRIE. Její pojmenování pochází z anglického „information retrieval“. Díky svým příznivým vlastnostem je často používána například při ukládání asociativních polí. Tato struktura existuje v několika provedeních. Další text se bude věnovat variantě „vícecestná TRIE“, která je vhodná pro uložení klíčů reprezentovaných řetězci.

6.1 Popis struktury

Jelikož se jedná o variantu vyhledávacího stromu, zaveďme pojem strom:

Definice 6.1 ***Kořenový strom*** *je acyklický graf, který má jeden zvláštní uzel, který nazýváme kořenem. Kořen je uzel, pro nějž platí, že z každého uzlu stromu vede jen jedna cesta do kořene. Z každého každého uzlu vede jen jedna hrana směrem ke kořeni do uzlu, kterému se říká rodičovský a libovolný počet hran k uzlům, kterým se říká synovské [2].*

Definice 6.2 ***List stromu*** *je uzel, který nemá synovské uzly.*

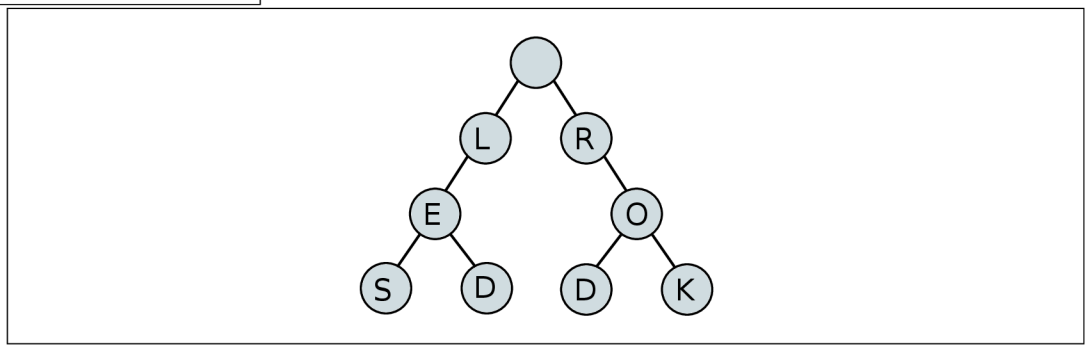
Definice 6.3 ***Vícecestný strom*** *je takový strom, v němž rodičovské uzly mohou mít více než dva synovské uzly.*

Definice 6.4 ***Vícecestné TRIE*** *je vícecestný strom, který má klíče přidruženy ke každému ze svých listů a je rekurzivně definován takto: TRIE pro prázdnou množinu klíčů je prázdný odkaz. TRIE pro jeden klíč je list obsahující tento klíč. TRIE pro množinu klíčů s mohutností větší než jedna je vnitřní uzel s odkazy ukazujícími do dalších TRIE pro klíče se všemi možnými hodnotami číslic s uvažovaným odstraněním vedoucí číslice pro účely vytváření podstromů [6].*

Jednotlivé klíče jsou uloženy v listech stromu, přičemž posloupnost hodnot rodičovských uzlů odpovídá hodnotám na začátku klíče. Proto se tato struktura také označuje jako prefixový strom.

Na diagramu 6.1 je znázorněné TRIE obsahující klíče *LES*, *LED*, *ROD* a *ROK*. První dva uvedené klíče mají společný prefix *LE*. Tento prefix je, podle definice 6.4, uložen v automatu jako společná část klíčů. Z posledního uzlu prefixu vedou hrany do uzlů tak, aby byly pokryty všechny klíče s tímto prefixem (tj. hrany se znaky ze 3. pozice v klíčích s prefixem *LE*, což jsou znaky *S* a *D*).

Diagram 6.1: TRIE



Rozdíl při vyhledávání pomocí TRIE a vyhledáváním pomocí jiných struktur je v tom, že se při porovnávání jednotlivých uzlů nepracuje s hodnotou celého klíče, ale jen s vybranou částí. Výběr části pro porovnání je závislý na vzdálenosti porovnávaného uzlu od kořene stromu.

Opět uvažujme TRIE z diagramu 6.1. Budeme vyhledávat klíč *ROD*. Začneme v kořenovém uzlu, ze kterého se přesuneme do synovského uzlu, který obsahuje první znak hledaného slova (tj. R). V novém uzlu postup opakujeme pro druhý znak (tj. O). Hledání prohlásíme za úspěšné, jestliže jsme v posledním kroku dosáhli listového uzlu a obsah tohoto uzlu se shoduje se zbytkem klíče, tedy řetězcem za posledně porovnávaným znakem.

Upozorněme na situaci, kdy je některý z klíčů prefixem jiného klíče. V tomto případě kratší z dvojice porušuje definici, že klíče jsou uloženy v listech. Proto je vhodné zajistit, aby se taková dvojice nemohla ve stromu vyskytnout. Jednou z možností je zavést pevnou délku klíče a ponecháním podmínky, že každý klíč musí být unikátní. Ze zřejmých důvodů není toto řešení příliš vhodné pro uložení klíčů, které reflektují slova z některého z přirozených jazyků.

Proto budeme využívat druhé možnosti, která spočívá v rozšíření klíče o speciální znak, který bude používán výhradně jako poslední znak klíče.

6.2 Souvislost s konečnými automaty

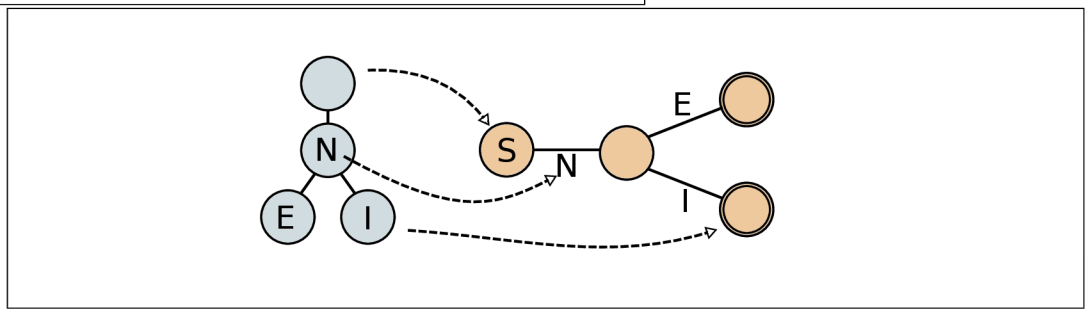
V této kapitole je konečným automatem myšlen nedeterministický konečný automat, podle definice z textu prof. Meduny [4].

Definice 6.5 ***Nedeterministický konečný automat*** *A je pětice $A = (Q, T, \delta, s, F)$*

kde: *Q je abeceda znaků*
T je abeceda vstupních symbolů
$\delta : Q \times T \to 2^Q$ je množina přechodů
$s \in Q$ je počáteční stav
$F \supseteq Q$ je množina koncových stavů

Stromovou strukturu lze pomocí jednoduchého algoritmu převést do reprezentace pomocí konečných automatů. Abychom mohli TRIE považovat za konečný automat, je nutné označit kořenový uzel za počáteční stav automatu. Dále musíme hodnoty jednotlivých uzlů přiřadit hranám do těchto uzlů směřujících. Hodnoty listových uzlů je nutné rozepsat jako

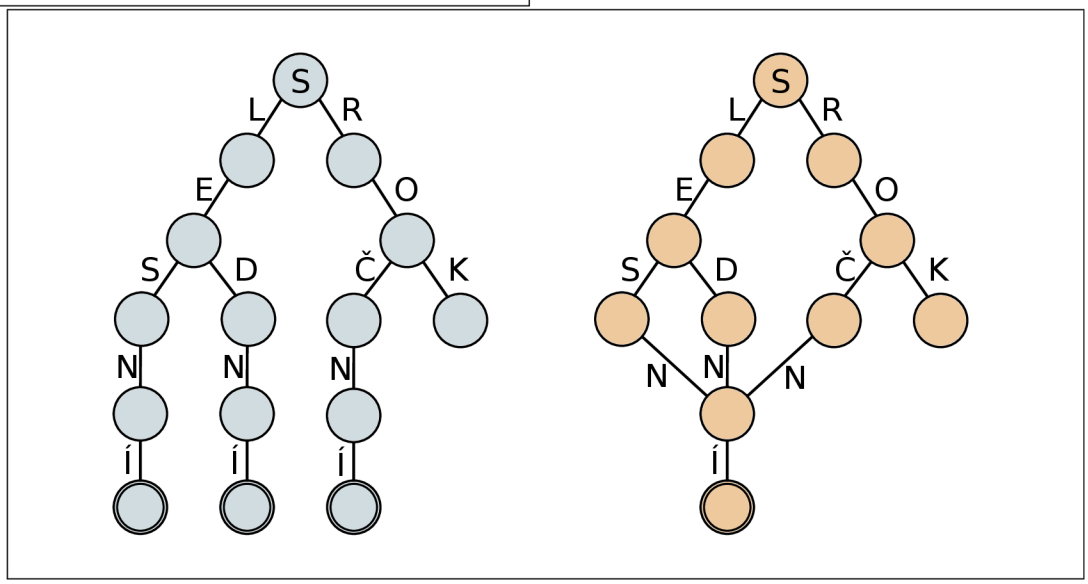
Diagram 6.2: Převod stromu na konečný automat



posloupnosti hran a stavů. Posledním krokem je označení koncových stavů. Jako koncový stav je označen každý stav, z kterého nevede alespoň jedna hrana.

Převedení TRIE ze stromové struktury do podoby konečného automatu není bezúčelné. Tímto krokem získáme větší volnost při ukládání slov. Velikou výhodou je možnost použití algoritmu pro minimalizaci konečných automatů. Díky této optimalizaci jsou zkomprimovány nejen předpony, ale i koncové části slov (viz diagram 6.3).

Diagram 6.3: FSA, Minimalizovaný FSA



Kapitola 7

Implementace

Pro uložení slovníku byl použit balík FSA, který slovník ukládá pomocí struktury TRIE převedené na konečné automaty.

7.1 FSA (Finite State Automata)

Autorem FSA je Jan Daciuk z Gdaňské univerzity. Balík obsahuje několik programů. Jedním z nejdůležitějších je fsa_build, který převede slovník z textového formátu do binárního. V textovém souboru je každý tvar slova na samostatném řádku. Spolu s tvarem slova je uloženo jeho lemma a anotace.

Lemma se zapisuje pomocí formátu označovaném *Kendings*. Tento formát je složen ze dvou částí. První označuje kolik písmen má být odebráno od konce tvaru. Druhou částí je text, kterým se nahradí odebraná část.

Formát anotace není balíkem omezen. V použitém slovníku byl využit formát popsaný v diplomové práci Radka Sedláčka [7]. Dále byla k anotaci přidána informace o vzoru, podle kterého je dané slovo skloňováno.

Funkci morfologického analyzátoru zastává program fsa_morph. Ostatní programy v balíku jsou určeny ke kontrole pravopisu, doplňování diakritických znamének, přibližnému určení morfologické analýzy, atd.

Celý balík je šířen pod svobodnou licencí GPL. Tato licence umožňuje použít zdrojové kódy s podmínkou, že výsledný produkt bude znovu šířen pod stejnou licencí.

Příklad 7.1: Zdrojová data pro vytvoření automatu

```
psí+A+k2eAgInPc5d1+vlčí
kočkou+Ca+k1gFnSc7+matka
```

Jednotlivá pole jsou oddělena znakem „+“. Pole označují v pořadí zleva tvar slova, *Kendings*, anotaci a vzor. Anotace druhého řádku označuje, že dané slovo je podstatné jméno, rodu ženského, čísla jednotného v 7. pádu.

7.2 Předpony

7.2.1 Uložení předpon v automatu

Velký význam pro použití budoucí aplikace má způsob uložení seznamu předpon. Tento seznam by bylo možné uložit do zvláštního souboru, který by se musel při analýze každého slova prohledávat, zda se některá z předpon uvedených v tomto souboru neshoduje s předponou analyzovaného slova.

Další možností je uložit předponu spolu s ostatními slovy do automatu. Pokud se při vyhledávání nalezne klíč odpovídající předponě, může se analyzátor pokusit zjistit, zda zbytek slova za předponou nepatří mezi slova, která lze spojit s danou předponou. Toto slovo se vyhledává stejně jako každé jiné, tedy začne se v kořenu. Předpona zde hraje úlohu jakéhosi skoku na začátek automatu. Touto rekurzí je i elegantně vyřešeno skládání více předpon.

Při uložení předpon spolu s ostatními slovy je nutné zajistit, aby analyzátor nezahrnul předpony mezi regulérní slova. Toho dosáhneme tím, že první část záznamu nebude od zbytku záznamu oddělena znakem „+“, tak jak je tomu u obyčejných slov, ale znakem „!“.

Z analýzy vyplývá, že předpony lze připojovat jen k některým slovům. Výběr těchto slov se provádí pomocí jejich anotace. Také je nutné mít prostředek pro úpravu anotace, pro případ že má přidání prefixu za následek její změnu. Tyto funkce zajišťují dvě anotace, které jsou v záznamu s předponou uloženy bezprostředně za znakem „!“.

Příklad 7.2: Záznam s předponou

```
ne!k2eAd1+eN
```

Tento záznam popisuje předponu *ne*. Tato předpona se může připojit k přídavnému jménu v 1. stupni, které není negací. Anotace uvedená za znakem „+“ označuje, že se slovo mění na negaci.

7.2.2 Manipulace s anotací

Při práci s anotací je využito toho, že každá informace je uložena v páru vlastnost, hodnota. Při výběru slov je za shodující se označeno jen takové slovo, u kterého odpovídají všechny páry informací uvedené v anotaci s anotací, podle které se porovnává. Pořadí jednotlivých párů neovlivňuje výsledek porovnání.

Úprava anotace slova probíhá opět podle anotace, která je uvedena v záznamu s předponou. Tato anotace je postupně procházena pár po páru. Každý z těchto párů upravuje jednu informaci ve výsledné anotaci.

7.2.3 Ošetření výjimek

V odstavci 3.1 jsou uvedeny některé z výjimek při skládání slova s předponou. Některé z těchto výjimek mohou být vysvětleny pomocí fonetiky[1]. K řešení využijme toho, že těchto slov, je mizivé procento z celkového počtu slov. Do slovníku uložme i správnou variantu s předponou a výchozí tvar označme jako slovo, ke kterému se předpony nemají připojovat. Tento přístup má i tu výhodu, že jej lze využít i pro další slova, ke kterým se nemají předpony vázat.

[1] Jedna z mnoha věd, která spolu s morfologií patří do lingvistiky. Tato věda se zabývá hláskami.

Příklad 7.3: Ošetření výjimky při tvoření negace ze slov *brát* a *nenávidět*

```
nebrat+A+k5eNaImF+brát
brát+A+!k5eAaImF+brát
```

Na prvním řádku je negace slova ve správné formě. Další řádek obsahuje původní tvar slova. Znak „!“ před anotací označuje, že se k tomuto tvaru nemá připojovat žádná předpona.

```
nenávidět+A+!k5eAaImF+závidět
```

Ačkoliv toto slovo vyjadřuje zápornou vlastnost, není negací. Aby mohl být tvar označen za negaci musí „negovat“ jiný tvar. Jelikož neexistuje něco jako *návidět*, nemůže být tento tvar označen za negaci.

Poslední speciální značkou se označuje předpona, která nemá být zahrnuta do lemma. Morfologický analyzátor předpony do lemma zahrnuje podle svého nastavení. Tato značka určuje, které přípony nemohou být do lemma zahrnuty v žádném případě.

Příklad 7.4: Vyjmutí předpony z lemma

```
nej!#k5d2+d3
```

Jelikož tato předpona jen upravuje tvar ve 2. stupni na tvar ve 3. stupni. Kdybychom tuto předponu zahrnuli, pak by výsledné lemma mohlo být například slovo *nejmladý*.

7.2.4 Analýza slov s předponou

Nejprve popišme postup analýzy slova bez předpony. Vyhledávání začíná od kořene slovníku. V prvním kroku označme kořen za výchozí uzel. V tomto uzlu vyhledáme hranu, označenou prvním znakem analyzovaného tvaru. Uzel, na který ukazuje tato hrana opět označme jako výchozí uzel a pokračujme vyhledáváním hrany, která je označena následujícím písmenem z analyzovaného slova. Takto pokračujeme, dokud nebudeme v uzlu, který odpovídá poslednímu znaku slova. Pokud hrana pro očekávaný znak neexistuje, pak toto slovo není ve slovníku uloženo a analýza končí.

Po nalezení posledního znaku slova je nutné zkontrolovat, zda z aktuálního uzlu vede hrana, která je označena speciálním znakem, který označuje konec klíče. Toto je nutné k ověření, zda jsme načetli celý klíč. Posloupnost následujících hran odpovídá anotaci nalezeného slova. Nutno poznamenat, že jeden tvar může mít několik anotací, které lze snadno získat některým z algoritmů pro průchod stromu.

Slovo s předponou se hledá obdobně. Jediná úprava algoritmu spočívá v tom, že se při vyhledávání hrany, odpovídající hledanému znaku, hledá zda z uzlu nevede hrana označující konec přípony (znak „!“). Pokud je tento znak nalezen, uloží se anotace, které určují, ke kterým slovům lze tuto předponu připojit a jak má být anotace těchto slov upravena.

Poté rekurzivně hledáme část slova za předponou. V případě, že nalezené slovo není označené jako takové, ke kterému se nemají připojovat přípony a zároveň se anotace nalezeného slova shoduje s první z anotací uložených u předpon, pak se provede úprava anotace. Pokud nebylo nalezeno žádné slovo, pak se nejedná o předponu a pokračuje se v analýze.

Správné skládání více předpon je ošetřeno tím, že se anotace upravuje postupně od nejpravější k nejlevější předponě. Tento postup přímo koresponduje s tvorbou slov pomocí přidávání předpon.

7.2.5 Porovnání velikosti

Použitím nové implementace významně klesl počet slov, který musí být uložen ve slovníku. Více než jedenáct milionů negací je uloženo pomocí deseti řádků. Obdobně je tomu u téměř čtyř milionů superlativů. Díky těmto úsporám se velikost slovníku zmenšila téměř na třetinu původní velikosti.

Velikost výsledného binárního souboru klesla pod desetinu původní velikosti. To je způsobeno tím, že se efektivněji aplikuje sloučení prefixů, ale především minimalizace automatu.

Tabulka 7.1: Parametry analyzátoru při použití různých implementací

Implementace	**Původní**	**S předponami**
Velikost slovníku [MB]	1 500,0	535,0
Velikost FSA [MB]	53,0	4,1
Rychlost [slov/s]	572 030,2	213 892,8

7.2.6 Test rychlosti

Obě implementace byly testovány v jedenácti experimentech. Výsledky byly ověřeny pomocí t–testu. Každý experiment spočíval v provedení analýzy téměř osmi milionů slov. Testy byly provedeny na systému s těmito parametry: CPU: Intel(R) Core(TM)2 Quad CPU Q6700 @ 2.66GHz; RAM: 4 GB; GNU/Linux (kernel 2.6.22).

Rychlost analýzy za použití nové implementace klesla na polovinu. Hodnota t z t–testu je 352,9. Toto zpomalení je způsobeno tím, že se musí vždy prohledávat všechny hrany vedoucí z uzlu. Seznam hran je seřazen sestupně podle počtu slov, na která daná hrana směřuje. Původní implementace může, bezprostředně po nalezení vyhovující hrany, přejít na další uzel. Nová implementace musí zkontrolovat, zda není jednou z hran i ta, která označuje předponu. Toto prohledávání je, zvláště v uzlech blízkých kořeni, důvodem takto výrazného zpomalení.

7.3 Číslovky

Při analýze číslovek je nutné vyřešit problém s abnormálním počtem slov. Uvažujme složeniny s číselným prefixem. Tvarů, ke kterým lze připojit číselný prefix je kolem dvaceti tisíc. Budeme-li chtít umět zanalyzovat všechna tato slova s prefixem majícím hodnotou do sto tisíc, získáme dvě miliardy možných tvarů. A to je jen část z celkového počtu číslovek. Proto je při analýze číslovek nutné zvolit jiný postup, než při analýze ostatních slovních druhů.

Místo ukládání všech tvarů, uložíme jen gramatiku, podle které se budou jednotlivé tvary kontrolovat.

7.3.1 Zápis číslovek

Zápis tokenů

Za tokeny jsou považovány všechny kořeny, interfixy a předpony, pomocí kterých se tvoří číslovky. U tokenů je nutné uchovávat hodnoty a operace, které reprezentuje. Dále je nutné uchovávat jednoznačné identifikátory jednotlivých tokenů. Kvůli zjednodušení analýzy, je nutné zajistit, aby žádný z tokenů nebyl prefixem jiného tokenu.

Příklad 7.5: Zápis tokenů

```
troj#b3d#+3
tisíc#bTa#*3
```

Záznamy jsou rozděleny znakem „#“. První část obsahuje kořen, ve druhé je uložen jeho identifikátor a v poslední části je uložena operace s hodnotou. Druhý řádek je obdobný. Operace na druhém řádku značí, že tento kořen násobí hodnotu mezivýsledku hodnotou 10^3.

Zápis koncových částí

Vedle tokenů je potřeba ukládat i seznam koncových částí. Do tohoto seznamu spadají jednak koncovky číslovek, ale také i části složenin vyskytující se za číselným prefixem. U koncových částí je nutné ukládat operace a unikátní identifikátory. Vlastní koncová část je ve stejném formátu jako záznam slova, které se ukládá do slovníku klasickým způsobem.

Příklad 7.6: Zápis koncový částí

```
; nula
c0:
  N#+Aa+k4gFnPc2+nula
  N#a+A+k4gFnSc1+nula
```

Tato část souboru označuje koncovky číslice *nula*. Znak „;“ označuje řádky s komentáři. Na následujícím řádku je identifikátoru skupiny koncovek. Název operace, která má být provedena je v záznamu uložena před znakem „#“. Za tímto znakem se nachází anotace jako u klasického slovníku, tedy znaky z konce slova, za kterými následuje *Kendings*, anotace a vzor.

Zápis gramatiky

V gramatice jsou uloženy všechny možné kombinace tokenů a koncových částí. Na tyto části se odkazuje výhradně pomocí jejich identifikátorů. Každý tvar je uložen na samostatném řádku. Pokud je tvar složen z více tokenů, jsou tyto tokeny odděleny tečkou. Identifikátor koncové části je od prefixu oddělen pomocí znaku „#“. Gramatika má i funkci pro změnu priority výpočtu hodnoty. Pokud je mezi kořeny znak „!“, pak má být hodnota mezivýsledku uložena do pomocné proměnné. Znak „@“ odděluje část, která reprezentuje hodnotu čitatele od zbytku zlomku.

Pro zjednodušení zápisu, lze skupinu tvarů označit jako non-terminál. Tato označení lze dále kombinovat mezi sebou i s jinými kořeny. Počáteční non-terminál je označen znakem „_“. Dále je uveden další non-terminál „+“, obsahující seznam koncových částí, které reprezentují číslovku i bez předpony.

Příklad 7.7: Zápis gramatiky

```
DVACET_a:
  b2e.sBa     ; dvacet
  b3c.sBa     ; třicet
  b4a.iia.sBa ; čtyřicet

; prefix pro složené číslovky tj. "jedenadvacet", apod.
JEDNOTKY_a:
  b1a.iaa       ; jedena
  b1e           ; jedna
  b2e.iaa       ; dvaa

; číslovky jedenadvacet, dvaatřicet, ...
DVAADVACET_a:
  JEDNOTKY_a!DVACET_a
```

Komentáře jsou označeny znakem „;“. Na posledním řádku je upravena posloupnost operací tak, aby se nejdříve vypočítaly desítky, ke kterým se následně přičte hodnota před nimi (`JEDNOTKY_a`).

7.3.2 Uložení v automatu

Výše popsané soubory jsou pomocí skriptu převedeny do tvaru, který je vhodný pro program fsa_build. Počáteční non-terminál je rozgenerován tak, že každý řádek obsahuje jen kořeny a koncové části.

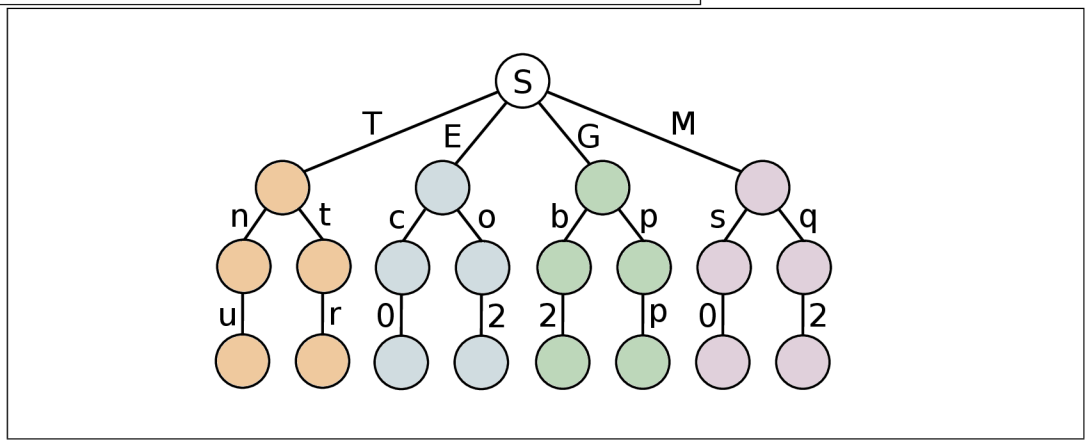
Pro zjednodušení manipulace s výsledným slovníkem, uložíme všechny tři seznamy do jednoho automatu. Každému ze seznamů přidáme unikátní prefix, čímž, díky rekurzivní definici TRIE rozdělíme automat do několika diskrétních částí (viz diagram 7.1).

Do části obsahující tokeny vede hrana T, do seznamu koncových částí vede hrana E. Gramatika je reprezentována hranou G, která ukazuje do vlastní gramatiky a hranou M, ve které je uložen seznam koncových částí, které mohou tvořit slovo i bez přidaného prefixu.

7.3.3 Zpracování číslovek

Číslovky jsou analyzovány na dvou úrovních. První část má za úkol rozdělení slova na tokeny. Druhá část má za úkol zkontrolovat, zda je posloupnost tokenů uvedena v gramatice. Dále má za úkol výpočet hodnoty číslovky. Použijeme-li termíny z konstrukce překladačů, pak budeme první část označovat za lexikální analýzu. Druhá část pak zahrnuje syntaktickou a sémantickou analýzu.

Diagram 7.1: Rozdělení automatu do diskrétních částí



Lexikální analýza

Tokeny se z analyzovaného slova separují tímto způsobem: Postupně se prochází automatem po hranách odpovídajících znakům z počátku slova. Jestliže budeme v uzlu, ze kterého vede hrana značící konec tokeny, pak byl token nalezen. Toto lze předpokládat díky tomu, že žádný token není prefixem jiného tokenu. Lexikální analýza vrací, buď chybu, to když se na začátku slova nevyskytuje žádný token. Při úspěchu se vrací identifikátor tokenu, jeho operace a zbytek analyzovaného slova po odebrání nalezeného tokenu.

Syntaktická a sémantická analýza

Tato část přímo využívá výsledků lexikální analýzy. V prvním kroku se zkontroluje, zda analyzované slovo nepatří mezi koncové části, které jsou zároveň slovy. Po této kontrole již nastává samotné zpracování gramatikou.

Pomocí lexikální analýzy se získá první token. Tento token se vyhledá v gramatice. Nebyl-li tento token v gramatice nalezen, pak analýza končí. Jestliže bylo hledání úspěšné, pak hrany vycházející z posledního uzlu ovlivňují další běh analyzátoru.

Je-li hrana označena znakem „.“, pak se pokračuje načtením dalšího tokenu. Jestliže je hrana označena znaky „@“ nebo „!“, pak se hodnoty mezivýsledku uloží do příslušných proměnných a opět se pokračuje načtením následujícího tokenu.

Hrana „#“ značí, že za ní následuje identifikátor konečných částí. Mezi těmito částmi se zkusí vyhledat analyzované slovo bez již nalezených tokenů. Při nalezení odpovídající koncové části, je analýza úspěšná a vrací morfologické informace uložené v koncovce a číselnou hodnotu číslovky.

Hodnota číslovky se vypočítává vždy po ověření, že nalezený token patří do gramatiky. Další úprava hodnoty probíhá při nalezení odpovídající koncové části.

7.3.4 Test rychlosti analýzy

Implementace zpracovaní číslovek, kterou popisuje tento dokument, je již druhou verzí. Testy v této kapitole porovnávají tyto dvě varianty.

Popis původní verze

V první verzi byly všechny části, tedy tokeny, koncovky a gramatika v separátních souborech, což zbytečně znesnadňovalo její použití. Gramatika obsahovala jen seznam kořenů. Změna priority výpočtu, či označení části slova za čitatel nebylo možné. Výpočet složených desítek byl realizován tak, že tvary desítek byly reprezentovány vlastními kořeny. To ovšem komplikovalo lexikální analýzu, protože prefixy těchto kořenů kolidovaly s kořeny jednotek.

Dalším problémem bylo, že nešlo přiřadit koncovky ke konkrétním tvarům přímo v gramatice. To bylo „obcházeno“ tím, že pro každou sadu koncovek musela být zvláštní gramatika a seznam tokenů. To opět velice komplikovalo analýzu, protože každé slovo muselo být testováno několika gramatikami.

Velký počet souborů byl částečně eliminován tím, že ty typy číslovek, jejichž počet to umožňoval, byly vygenerovány do klasického slovníku. Tento slovník byl zhruba stejné velikosti jako slovník pro zbylé slovní druhy. I při použití tohoto slovníku bylo stále nutné analyzovat každé slovo několika gramatikami.

Všechny tyto nepříjemné vlastnosti byly ve druhé verzi odstraněny.

Tabulka 7.2: Porovnání počtu a velikosti souborů

	Počet souborů	**Celková velikost souborů**
Původní implementace	7	2,4 MB
Nová implementace	1	51 KB

Způsob testování

Testování probíhalo na stejném systému jako testování předpon (viz kapitola 7.2.6). Výsledky jsou aritmetickým průměrem jedenácti experimentů. Jako testovací data byly použity tyto tři soubory:

words.txt – seznam slov ze všech slovních druhů, vyjímaje číslovky. Soubor obsahuje 3 959 306 slov.

numbers.txt – seznam obsahující jen číslovky. Soubor obsahuje 1 000 000 slov.

merge.txt – sloučení výše uvedených slovníků.

Tabulka 7.3: Porovnání rychlosti v závislosti na druhu analyzovaných slov

	words.txt	**numbers.txt**	**merge.txt**
Původní implementace [slov/s]	191 624, 75	85 884, 64	238 575, 87
Nová implementace [slov/s]	221 656, 54	75 669, 28	278 420, 67

Tabulka 7.3 ukazuje, že nová implementace je rychlejší na souboru slov, které patří převážně do jiného slovního druhu než do číslovek. Analýza číslovek je mírně pomalejší, což při normální skladbě slov nezpůsobí velké zpomalení.

7.4 Slovotvorné vazby

Pro generování slovotvorných vazeb byl zvolen postup, kdy je na fundující slovo aplikována množina vazeb.

7.4.1 Uložení vazeb

Vazby nesou hned několik informací, proto bylo nutné navrhnout vhodný formát pro jejich zadávání.

Z kmene mohou vést jen dva typy vazeb a to vazby s předponou nebo příponou. Předpony lze na kmen aplikovat bez dalších jeho úprav. U přípon je nutno před jejich použitím odstranit koncovku. Každá vazba dále nese sémantickou informaci. Pokud vazba vytvoří tvar, ze kterého lze odvozovat další tvary, je nutné specifikovat množinu vazeb, které mohou být na tvar aplikovány.

Příklad 7.8: Uložení vazeb

```
Dělat_A:
        = Dělat
        @ pře, P01: Dělat
        @ u, P02: Dělat
        @ vz, P03: Vzdělat

Dělat:
        & 2 ání, Sub21
        & 1 ný, Adj98 :Dělaný

Dělaný:
        & 1 ě, Adv00
```

Řádek začínající znakem „=“ označuje, že se na daný tvar má aplikovat množina vazeb, jejíž označení je uloženo za tímto znakem. Řádky začínající znakem „@“, reprezentují hranu s předponou. Řádky s příponami začínají znakem „&“. Za tímto úvodním znakem jsou uloženy informace, pomocí kterých se upraví tvar. Číslo se uvádí jen u vazby s příponou. Toto číslo uvádí, kolik znaků má být odstraněno z konce tvaru. Dále je uvedena předpona nebo přípona, která má být ke slovu připojena.

Za znakem „,“ je uvedeno označení hrany, toto označení může být použito k uložení sémantiky. Za znakem dvojtečky je uvedeno označení množiny vazeb, která má být na tvar aplikována. Pokud se na řádku nevyskytuje dvojtečka, pak se jedná o koncový tvar (tj. již se z něj neodvozují další tvary).

7.4.2 Uložení kořenů

Ke každému kmenu je uložena množina vazeb, pomocí kterých se odvozují další tvary hnízda.

Příklad 7.9: Uložení kořenů

```
děl: Děl
```

Před znakem dvojtečky je uložen tvar kořene. Za dvojtečkou se nachází označení množiny vazeb, která se bude na kmen aplikovat.

7.4.3 Uložení v automatu

Do automatu se ukládají vždy dvojice slov. První z nich je slovo odvozené, druhým je fundující slovo. Spolu s těmito slovy je uložena i vazba, která obě slova spojuje. Fundující slovo je uloženo pomocí *Kendings*. Pro uložení slov, která byla odvozena pomocí prefixace, byla navrhnuta obdoba *Kendings*. V této variantě se ukládá jen počet písmen, které mají být odstraněny ze začátku slova. Od původní varianty je odstraňování prefixu rozlišeno tím, že je číselná hodnota reprezentována malým písmenem a nikoliv velkým.

7.4.4 Vyhledání analýzy slova

Každé slovo má v automatu uložen tvar, ze kterého bylo slovo odvozeno. S tímto tvarem je uložena i sémantická informace o vazbě. Vyhledáním slova v automatu získáme slovo, které je v hierarchii blíže ke kmeni. Dále tímto získáme sémantickou informaci. Toto hledání se opakuje vždy se získaným (fundujícím) slovem. Hledání končí u slova, ke kterému již nelze získat výchozí slovo, protože nalezené slovo je kořenem.

7.5 Knihovna libma

V této kapitole je jen nástin funkcionality knihovny. Na přiloženém datovém médiu je spolu s knihovnou uložena i její podrobná dokumentace.

Tato knihovna zapouzdřuje všechny funkce, které byly popsány v tomto dokumentu, do jednoho rozhraní. Knihovna je napsána v jazyce C++. Na rozdíl od výkonných funkcí, které používají vlastní, optimalizované datové typy, je rozhraní knihovny založeno jen na typech z STL[2]. Díky tomu je pro nového uživatele knihovny snadnější ji začít využívat. Bohužel je nutné uvést, že použití standardních typů má za následek zpomalení analýzy, před kterou musí být tyto typy převedeny na typy využívané výkonnými funkcemi. Dalším prostředkem, kterým je uživateli ulehčen přechod na knihovnu, jsou okomentované demonstrační příklady.

Před vlastní prací s knihovnou je nutné vybrat datové soubory se slovníky. Seznam těchto souborů je uveden v konfiguračním souboru. Tento soubor slouží i pro nastavení jiných vlastností analyzátoru, jako je třeba úroveň lemmatizace či citlivost na velikost písmen.

Analyzátor je reprezentován třídou *ma_fsa*, jejíž konstruktor očekává jeden parametr s cestou ke konfiguračnímu souboru. Vlastnosti analyzátoru lze nastavovat pomocí Get/Set rozhraní.

Vlastní morfologickou analýzu provádí funkce *Morph*, které je předáno slovo určené k analýze a vektor řetězců, do kterého má být výsledná analýza uložena. Pro získání číselné hodnoty slouží funkce *GetValue*, která vrací hodnotu posledně analyzovaného slova. Funkce vrací hodnotu NAN, jestliže posledně analyzované slovo nereprezentovalo číselnou hodnotu.

[2]Standard Template Library

V adresáři pylibma je uložen modul s rozhraním pro jazyk Python. Toto rozhraní se nechová zcela přesně jako originální knihovna, protože se snaží zachovat princip nejmenšího překvapení, který se v tomto jazyce uplatňuje.

Knihovna byla testována na operačním systému GNU/Linux a FreeBSD. Jiné systémy testovány nebyly. Celá knihovna je šířena pod svobodnou licencí GPL.

Příklad 7.10: Konfigurační soubor pro knihovnu libma

```
# Language file
language = "lang/cz.lang"

# Level of base form
# 0 - without prefixes
# 1 - with prefixes
lemmatization = 0

# Letters case
# 0 - exactly same as word in dictionary
# 1 - first letter could be changed to upper case
# 2 - all of above and capitalized word will be accepted too
# 3 - case of letter doesn't matter
case = 3

dictionary = "data/prijmeni.fsa"
dictionary = "data/czech_comp.fsa"

numbers = "data/numeral.fsa"
```

Proměnná „language“ obsahuje cestu k souboru, který popisuje jednotlivá písmena češtiny. Tento popis je využit při převádění velkých a malých písmen s diakritikou. Další proměnná nastavuje, zda se bude za základní tvar požadovat slovo spolu s předponou nebo slovo bez předpony. Proměnná „case“ nastavuje citlivost k velikosti písmen. Hodnotou 3 je nastaveno, že slova budou přijata bez ohledu na velikost jejich písmen.

Poté již následují dvě datové proměnné, kterými se vybírají klasické slovníky. Slovník s gramatikou číslovek je vybrán na posledním řádku.

Kapitola 8

Závěr

V předchozích kapitolách bylo předvedeno, jak lze speciálním uložením předpon razantně zmenšit objem dat, který je nutný k uložení slovníku s morfologickou anotací.

Aby byla co nejvíce využita rychlost vyhledávání v klasické TRIE, byly speciálně uloženy jen nejfrekventovanější předpony, které se navíc na slova aplikují, až na pár výjimek, pravidelně, čímž odpadla potřeba udržovat seznamy slov a předpon, které lze spolu slučovat. Výsledná implementace je schopna uložit shodnou množinu tvarů na zhruba desetkrát menším prostoru.

Pro analýzu číslovek byl navrhnut silný aparát, kterým lze obsáhnout všechny druhy číslovek a co víc, získat z nich číselnou hodnotu. Nynější systém umí analyzovat několik miliard tvarů číslovek, ale i přes tento počet je stále nutné doplňovat „slovník“ méně častými tvary přidávat, aby se dosáhlo co největšího pokrytí tohoto slovního druhu.

V budoucnosti bude práce na analyzátoru pokračovat. Zejména se doplní definice slovotvorných vazeb tak, aby byla i tato část analyzátoru použitelná v praxi. Implementace pro generování vazeb bude doplněna dvouúrovňovou morfologií. První úroveň bude zajišťovat generování nových tvaru, zatímco druhá úroveň bude výsledné tvary upravovat podle zásad fonologie.

Do analyzátoru budou přidány funkce, které budou schopny ze zadaného tvaru generovat všechny možné tvary tohoto slova nebo vracet tvar podle zvolené morfologické kategorie.

Literatura

[1] Bohuslav HAVRÁNEK and Alois JEDLIČKA. *Česká mluvnice.* Státní pedagogické nakladatelství, SPN Praha, 1951.

[2] Jan M. HONZÍK. *Algoritmy.* Studijní opora, FIT VUT v Brně. Brno, 2007.

[3] Lexikografický kolektiv ÚJČ AV ČR. *Slovník spisovné češtiny.* Academia, Praha, 4 edition, 2007.

[4] A. MEDUNA and R. LUKÁŠ. *Formální jazyky a překladače.* Studijní opora, FIT VUT v Brně. Brno, 2006.

[5] Karel PALA, Radek SEDLÁČEK, and Marek VEBER. Vztah mezi tvarotvornými a slovotvornými vzory v češtině. In *Čeština – univerzália a specifika*, pages 151–162. Nakladatelství Lidové noviny, Praha, 2004. Masarykova univerzita v Brně.

[6] Robert SEDGEWICK. *Algoritmy v C.* Addison-Wesley Publishing Company, Inc, 1998. ISBN 80-86497-56-9.

[7] Radek SEDLÁČEK. Morfologický analyzátor češtiny. Master's thesis, FI MU v Brně, Brno, 1999.

[8] Radek SEDLÁČEK. *Morphemic Analyser for Czech.* PhD thesis, FI MU in Brno, Brno, 2004.