Fast Morphological Analysis of Czech

Pavel Šmerk, FI MU

- motivace: Seznam.cz, IS MU/theses/... (fulltext, plagiáty)
- využití DAFSA Jana Daciuka
- jazyk automatu je seznam dotaz:odpověď:

```
klouček:A:k1gMnSc1
kloučka:Cek:k1gMnSc2
kloučka:Cek:k1gMnSc4
```

- tedy žádná analýza ve smyslu počítání, pouze průchod automatem a výpis všech možných pokračování cesty
- tedy rychlost při běhu a jednoduchost kódu

Srovnání s ajkou

	velikost v MB		čas		
	ajka	majka	ajka	majka	poměr
analýza	3.1	4.4	18.22	2.88	**6.3x**
lemmatizace		4.0	16.76	1.57	**10.7x**
generování tvarů		6.1	55.33	8.42	**6.6x**
diakritika		3.3	8698.8	1.61	**5403x**

- prvních 1M pozic ze SYN2000
- průměr ze tří časů, celkový čas z příkazu time
- výstup přesměrován do /dev/null
- drobné rozdíly ve složeninách, které by neměly mít vliv
- data celkem větší, ale zaráz potřebuju jen jeden slovník

Data

slovník	položek	velikost dat	automatu	B/pol.
w	13 609 590	186 154 068	3 263 374	0.240
w → l	14 101 767	239 578 702	4 042 839	0.287
w → lt	80 303 929	2 477 786 062	4 353 616	0.054
w → w	957 464 060	19 993 465 213	6 105 429	0.006

- + l-w, l-wt, w-wt, lt-w, wt-w (nejde přímo srovnat s ajkou)
- počty položek zahrnují i gramatiku
- brazilská portugaština 0.25, němčina w-lt 0.15
  - je to ovšem blbost, informace je tam holt pár MB ;-)
  - zřetelně ale obava z velkého seznamu není důvodná
  - (w-wt jsem zrušil při cca 0.25 TB, teď už snad je)

- Relativně k /nlp/projekty/ajka:
- `echo test | bin/majka -f lib/majka.w-lt`

```
test:k1gInSc1
test:k1gInSc4
test:k1gMnSc1
testa:k1gFnPc2
```

- knihovna lib/libmajka.a + lib/majka.h, příklad lib/majka.cc
- po inicializaci k dispozici 4 metody
  - maxvelikost(), najdi(word, buff), ohackuj(word, buff), vysledku
  - Yenya chce najdi + najdi_dalsi, thread-safety
- data pro synt
  - zachovávají stupeň a negaci, bez blacklistu, wH, m[SD], kA, xy u k3468, za ~ jsou 1 + trunc(log_3.1(frekvence z desamu))

- perlové rozhraní:

```
perl -I/nlp/projekty/ajka/lib -Mmajka -e '
   $m = majka::automat->new("
         /nlp/projekty/ajka/lib/majka.w-lt");
   print map "$_\n", @{$m->find("test")}'
test:k1gInSc1
test:k1gInSc4
test:k1gMnSc1
testa:k1gFnPc2
```

- rozhraní pro Python by musel někdo udělat
  - Perl to čistý swig, musel jsem to krapet hacknout
  - nebo až dodělám najdi + najdi_dalsi

- stále ve vývoji ;-)
  - očekávám zrychlení, zmenšení dat i kódu
  - aktuálně lžu o potřebné velikosti bufferu
- kód je 5x kratší
  - ovšem teda včetně komentářů, mrtvého kódu atp.
  - podstatné je, že kód nafukuje zpracování nalezených dat, pro každý automat jiné, vzájemně ale nezávislé
  - Radek implementoval allt, a neuspěl...
- uvnitř je to v il2, je to o fous rychlejší
- lze pod win (zatím bez cp1250 a \r)

- Budoucnost
- zkusit pořadí značka/lemma, ve značce (zmenšení samo o sobě ne až tak zajímavé, možná spíš pro cache)
- pořádnou gramatiku
- (vzory, derivace, ...)

- Kdybychom se nudili... ;-)
- w-w generování:
  - Alice:A, Alice:Bi, Alice:Bí X Alice:Be, Alice:Bi, Alice:Bí
- fsa v derivu (ovšem judy?)
- magické 5% zpomalení
- měření rychlosti pro lemmata: asi nutno náhodně ;-)