

Posudek bakalářské práce

předložené na Matematicko-fyzikální fakultě
Univerzity Karlovy v Praze

posudek vedoucího

posudek oponenta

Autor/ka: DAVID BARČEK

Název práce: NOVELIZACE ČÁKUNŮ

Studijní program a obor: INFORMATIKA – OBECNÁ INFORMATIKA

Rok odevzdání: 2006

Jméno a příjmení vedoucího/opponenta: MGR. DAVID KLUSÁČEK

Pracoviště: ÚPAL

	excelentní	velmi dobrá	dobrá	nevýhovující
Obsádnost zpracovaného tématu	X			
Obsah odborní znalosti	X			
Struktura textové části práce	X			
Formát a typografická úprava		X		
Angličtina		X		
Učebně odborná dokumentace		X		
Literaturně odborná dokumentace		X		
Kvalita zpracování softwarové části			X	
Stabilita aplikace	X			

Nejvýznamnější klady:

Student si vybral pomerne komplikovane tema, ktereho se velmi dobre zhostil. Budu-li v dalsim textu kriticky, je treba to brat relativne. Celkove jsem totiz spokojen.

Libil se mi spravny pristup pri testovani programu, kde autor porovnaval vygenerovane novely s jejich skutecnym znenim, presto ze to prinaselo pracne problemy s rucni upravou zakonu (jelikoz ty obsahuji nekttere informace, kttere neplynou z novel – napriklad data prechodnych ustanoveni vztahujicich se k dannemu zakonu se lisi). Tento test ukazal ze pres 95% vygenerovanych vet se nelisi od textu zakona. Je tez treba ocenit velkou pracnost pri ziskavani textu zakonu a novel a jejich prevodu do standardniho tvaru.

Ocenuji pouziti systemu TeX pro sazbu prace. Vyznamnym kladem prace je tez její strucnost.

Nejzávažnější nedostatky:

Za nejvetsi nedostatek lze oznacit pouziti nevhodneho parsovacího algoritmu (viz. str. 20), kttery ma exponencialni casovou slozitosť, presto ze je znam algoritmus polynomialni. Prakticke pouzitelnosti to sice nevadi, ale stane se, ze nekttere vety program zpracovava i dele nez sekundu, coz je mnoho, kdyz uvazime jednoduchosť gramatiky.

Dalsi progrese vidim ve zpusobu naprogramovani. Pokud se autor moc nekararadi s C++, mel si zvolit radeji nejaky jiny jazyk ve kterem lze k problemu bez obav pristupovat na vyssi urovni abstrakce (napr. Haskell nebo Mercury). Vyhnul by se tak trapnemu pouzivani retezcu misto vycetovych typu, podivnemu rozvrzeni parseru na jednotlivé objekty apod. Toto se v C++ nedela:

```
if (nazev == "retezec") promenne.push_back(new Typ_Retezec(parametry[0]));
else if (nazev == "vycet") promenne.push_back(new Typ_Vycet(parametry[0]));
else if (nazev == "vec") promenne.push_back(new Typ_Vec(parametry[0]));
else if (nazev == "misto") promenne.push_back(new Typ_Misto(parametry[0]));
else if (nazev == "skupina") promenne.push_back(new Typ_Skupina(parametry[0]));
else if (nazev == "veta") promenne.push_back(new Typ_Veta(parametry[0]));
else if (nazev == "souveti") promenne.push_back(new Typ_Souveti(parametry[0]));
```

Podotykam ze nazev je typu string takze == vlastne provadi strcmp().

Další poznámky:

Autor se prilis nezabyva otazkou co delat, kdyz je gramatika viceznacna a parser by mohl vratit vice stromu – zminka je na str. 8. Ve skutecnosti je to ale zajimavy problem: bylo by napriklad mozne nekttere z nich zahrnout protoze neuspeje prislusne vyhledani pasaze v zakone zatimco u jinych stromu ano; u tech co zbydou by bylo mozne zmeny provest a pak vysledne texty ohodnotit statistickym modelem natrenovanim na textech zakonu a vybrat nejpravdepodobnejsi variantu).

Jeste nekttere detaily: Typografie – lepsim dojmem pusobi \parskip=0pt a pouzivani \noindent v odstavci tesne za nadpisem. Styl – odkazovat se v budoucim case z kapitoly 3.5 do 3.4 zni divne. Styl neni prilis formalni coz muze byt obcas na ukor presnosti vyjadrovani.

	výborně	velmi dobře	dobře	něprospěš/a
Návrh známky	⊗			

Datum: 16.6.2006

Podpis:

