

Primena mašinskog učenja u verifikaciji softvera

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Nikola Dimitrijević, Rastko Đorđević,
Luka Živanović, Dimitrije Špadijer
nikoladim95@gmail.com, mi14078@alas.matf.bg.ac.rs,
mi14164@alas.matf.bg.ac.rs, mm11021@alas.matf.bg.ac.rs

31. mart 2018

Sažetak

U ovom tekstu je ukratko prikazana osnovna forma seminarskog rada. Obratite pažnju da je pored ove .pdf datoteke, u prilogu i odgovarajuća .tex datoteka, kao i .bib datoteka korišćena za generisanje literature. Na prvoj strani seminarskog rada su naslov, apstrakt i sadržaj, i to sve mora da stane na prvu stranu! Kako bi Vaš seminarski zadovoljio standarde i očekivanja, koristite uputstva i materijale sa predavanja na temu pisanja seminarskih radova. Ovo je samo šablon koji se odnosi na fizički izgled seminarskog rada (šablon koji *morate* da ispoštujete!) kao i par tehničkih pomoćnih uputstava. Molim Vas da kada budete predavali seminarski rad, imenujete datoteke tako da sadrže temu seminarskog rada, kao i imena i prezimena članova grupe (ili samo temu i prezimena, ukoliko je sa imenima predugačko). Predaja seminarskih radova biće isključivo preko web forme, a NE slanjem mejla.

Sadržaj

1	Uvod	2
2	Verifikacija softvera	2
2.1	Dinamička verifikacija softvera	3
2.2	Statička verifikacija softvera	3
3	Drugi naslov	3
3.1	... podnaslov	3
4	n-ti naslov	3
4.1	... podnaslov	3
4.2	... podnaslov	3
5	Poslednji naslov	4
6	Zaključak	4
	Literatura	4

1 Uvod

Ko želi, može da piše rad ћирилицом. У том случају, неопходно је да су инсталирани одговарајући пакети: `texlive-fonts-extra`, `texlive-latex-extra`, `texlive-lang-cyrillic`, `texlive-lang-other`.

Уз све нововедене termine u zagradi naglasiti od koje engleske reči termin potiče. Naredni primeri ilustruju način uvođenja enlegskih termina kao i citiranje.

Primer 1.1 *Problem zaustavljanja (eng. halting problem) je neodlučiv [3].*

Primer 1.2 *Za prevođenje programa napisanih u programskom jeziku C može se koristiti GCC kompajler [1].*

Primer 1.3 *Da bi se ispitivala ispravost softvera, najpre je potrebno precizno definisati njegovo ponašanje [2].*

Reference koje se koriste u ovom tekstu zadate su u datoteci *seminarski.bib*. Prevođenje u pdf format u Linux okruženju može se uraditi na sledeći način:

```
pdflatex TemaImePrezime.tex
bibtex TemaImePrezime.aux
pdflatex TemaImePrezime.tex
pdflatex TemaImePrezime.tex
```

Prvo latexovanje je neophodno da bi se generisao *.aux* fajl. *bibtex* proizvodi odgovarajući *.bbl* fajl koji se koristi za generisanje literature. Potrebna su dva prolaza (dva puta `pdflatex`) da bi se reference ubacile u tekst (tj da ne bi ostali znakovi pitanja umesto referenci). Dodavanjem novih referenci potrebno je ponoviti ceo postupak.

Broj naslova i podnaslova je proizvoljan. Neophodni su samo Uvod i Zaključak. Na poglavlja unutar teksta referisati se po potrebi.

Primer 1.4 *U odeljku ?? precizirani su osnovni pojmovi, dok su zaključci dati u odeljku 6.*

Još jednom da napomenem da nema razloga da pišete:

```
\v{s} i \v{c} i \'c ...
```

Možete koristiti srpska slova

```
š i č i ć ...
```

Ovde pišem uvodni tekst. Ovde pišem uvodni tekst. Ovde pišem uvodni tekst. Ovde pišem uvodni tekst.

2 Verifikacija softvera

U savremenom dobu, računari i računarski sistemi predstavljaju sastavni deo svakodnevice, kako u privatnom životu, tako i u poslovnom svetu, a i u državnoj administraciji. Zato je od izuzetnog značaja da softver koji se koristi bude pouzdan. Neispravan softver, u zavisnosti od toga gde se koristio i koliko je veliki bio propust, može izazvati male, ali i ogromne probleme sa teškim posledicama (čak i smrtnim). Oblast razvoja softvera koja se bavi proverom ispravnosti softvera, odnosno potvrđivanjem da softver radi u skladu sa zahtevima, naziva se *verifikacija softvera*.

Potrebno je precizirati šta se podrazumeva pod ispravnim softverom. Razlikujemo potpuno ispravan i delimično ispravan softver. Softver se smatra potpuno ispravnim ako se zaustavlja za svaki ulaz i na izlazu daje ispravan rezultat, dok se delimično ispravnim softverom smatra onaj koji za ulaz daje ispravan rezultat ako da se zaustavlja, a dozvoljeno je da se za neki ulaz program ne zaustavlja. Ne postoji algoritamski način da se proverí da li se neki program zaustavlja (tzv. Halting problem), pa se često ni ne ispituje potpuna, već samo delimična ispravnost softvera.

Postoje dve osnovne vrste verifikacije softvera. To su dinamička i statička verifikacija softvera.

2.1 Dinamička verifikacija softvera

Dinamička verifikacija softvera zasnovana je na tome da se provera ispravnosti softvera vrši tokom njegovog izvršavanja. Najčešće se dinamička verifikacija softvera vrši testiranjem i ti pojmovi se poistovećuju, što nije sasvim ispravno.

Testiranje je složen proces koji obuhvata pronalaženje što raznovrsnijeg skupa ulaza, definisanje očekivanih izlaza za svaki od tih ulaza, a zatim izvršavanje programa i provera da li je program za date ulaze vratio odgovarajuće izlaze.

2.2 Statička verifikacija softvera

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

3 Drugi naslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

3.1 ... podnaslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

4 n -ti naslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

4.1 ... podnaslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

4.2 ... podnaslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

5 Poslednji naslov

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

6 Zaključak

Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak.

Literatura

- [1] Free Software Foundation. GNU gcc, 2013. on-line at: <http://gcc.gnu.org/>.
- [2] J. Laski and W. Stanley. *Software Verification and Analysis*. Springer-Verlag, London, 2009.
- [3] A. M. Turing. On Computable Numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.