

Programiranje bazirano na invarijantama

Seminarski rad u okviru kursa
Verifikacija softvera
Matematički fakultet

Stefan Zarić
mi12147@alas.matf.bg.ac.rs

14. maj 2018.

Sažetak

U ovom seminarskom radu su opisani osnovni pojmovi programiranja baziranog na invarijantama, prikazani su dijagrami invarijanti kao osnovni koncept ovog načina programiranja. Zatim se objašnjava postupak konstrukcije programa baziranih na invarijantama kao i način za proveru njihove korektnosti. Nakon toga su navedene neke razlike između invarijantnih programa i uobičajenih strukturalnih programa. Na samom kraju izneta su iskustva iz predavanja kurseva iz invariantnog programiranja.

Sadržaj

1	Uvod	2
2	Dijagrami invarijanti	2
3	Konstrukcija programa baziranih na invarijantama	2
4	Provera korektnosti	3
5	Razlike u odnosu na uobičajene programe	3
6	Iskustva	4
7	Zaključak	4

1 Uvod

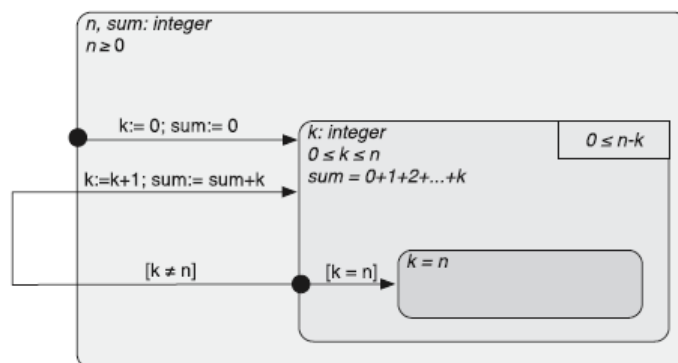
Osnovna ideja programiranja baziranog na invarijantama koje je predstavio Ralph-Johan Back u svom radu *Invariant based programming: basic approach and teaching experiences*, je da se specifikacije i invarijante petlji definišu pre samog pisanja koda. Na taj način verifikacija postaje dodatak fazi pisanja koda a ne testiranja, čime dobijamo korektnost programa pri samom pisanju istog. Verifikacija programa se obično radi u sledećim koracima:

1. Definisanje preduslova i postuslova
2. Definisanje invarijanti petlji
3. Izračunavanje uslova verifikacije
4. Provera da su svi uslovi verifikacije ispunjeni

Razlika u odnosu na uobičajen način programiranja je što se uobičajeno prvo piše kod pa se definišu preduslovi, postuslovi, invarijante i onda radi provera uslova, dok se kod invarijantnog programiranja prvo definišu preduslovi i postuslovi zatim se definišu invarijante petlji a nakon toga piše kod i radi provera uslova. Takođe moguće je u svakoj fazi vratiti se u bilo koju prethodnu ukoliko treba nešto ispraviti.

2 Dijagrami invarijanti

Dijagrami invarijanti su slični dijagramima stanja, ali dijagrami stanja samo prikazuju kontrolu toka programa ne razmatrajući korektnosti, dok dijagrami invarijanti zahtevaju korektnost u svakom stanju programa.



Slika 1: Dijagram invarijanti

Na slici 1 prikazan je primer dijagrama invarijanti za jednostavan program za sumiranje prvih n prirodnih brojeva.

3 Konstrukcija programa baziranih na invarijantama

Konstrukcija invarijantnih programa se radi tako što se na početku precizno definiše problem koji treba rešiti, zatim se definiše početni položaj u

kome se program nalazi i svi uslovi koji moraju biti ispunjeni kao i stanja promenljivih, i definiše se krajnji položaj odnosno stanja promenljivih na kraju izvršavanja programa. Nakon toga se pronalazi neki srednji položaj i definiše se invarijanta za taj položaj, kao i prelazi i početnog položaja u srednji koji se povezuju usmerenom linijom na dijagramu invarijanti sa definisanim promenljivama već definisanim u početnom položaju koje se mogu izmeniti u srednjem. Takođe se u gornjem desnom uglu srednjeg položaja prikazuje uslov za izlazak iz petlje tj. tog položaja. Zatim se koliko god je potrebno definišu novi položaji da bi se definisao algoritam za dolazak u stanje definisano srednjim odnosno svakim novim položajem. Na taj način pravi se ugnježden dijagram invarijanti gde je precizno definisano koji sve uslovi treba da budu ispunjeni u svakom stanju programa, čime ukoliko je program ispravno konstruisan dobijamo program koji je korektan u svakom njegovom stanju. Tako konstruisan program se može prevesti u neki od viših programskih jezika ili direktno u assembler da bi se izvršio.

4 Provera korektnosti

Da bi program bio korektan dijagram invarijanti treba da ispunjava određene uslove. Dijagram treba da bude konzistentan, da se završava i da nije blokirajući odnosno da će izvršavanje programa u nekom trenutku stići do krajnjeg položaja. Za proveru korektnosti se koriste alati koji automatski proveravaju dijagrame invarijanti. Na primer Socos system je alat koji se koristi za grafičko formiranje dijagrama invarijanti čiji izlaz se može poslati nekom automatskom ili interaktivnom dokazivaču.

5 Razlike u odnosu na uobičajene programe

Razlika između uobičajene programe je u samom pristupu problemu koji se rešava. Uobičajene programe smatramo korektnim ako program koji je počeo da se izvršava u početnom položaju i ispunjeni su preduslovi, nakon izvršavanja budu ispunjeni postuslovi. Dok kod invarijantnih programa svaki središnji položaj može biti početni i time je program korektan u svakom stanju i početni uslovi su ispunjeni u svakom položaju u kome se program može naći tokom izvršavanja.

Uobičajeni programi se pišu tako da taj kod kasnije mora biti testiran i debugovan dok invarijantni programi se pišu tako da već u fazi pisanja imamo definisane sve invarijante i korektnost programa proističe iz samog takvog načina programiranja. Potrebno je uložiti dodatni trud u fazi pisanja programa ali dobijamo veću pouzdanost tako napisanih programa u odnosu na uobičajene.

Kao što je već rečeno invarijantni programi se konstruišu tako što se prave dijagrami invarijanti i samim tim invarijante će biti dostupne u svakom sledećem koraku pri konstrukciji programa.

Invarijantni programi se fokusiraju na manje jedinice posla u odnosu na uobičajene programe. Svaki problem se može posmatrati lokalno jer se svaki položaj rešava za sebe pri čemu se garantuje korektnosti pri prelascima iz jednog položaja u drugi.

6 Iskustva

Autori rada koji opisujemo su organizovali više sesija na kojima je predstavljeno invarijantno programiranje i kroz vežbe sa primerima i rešavanjem nekih poznatih programerskih problema hteli su da dođu do zaključka o tome da li je invarijantno programiranje stvarno primenljivo u praksi. Takođe su održana dva kursa o invarijantnom programiranju, jedan je držan studentima na doktorskim studijama, a drugi je držan studentima prve godine osnovnih studija.

U organizovanim sesijama su programeri rešavali probleme u paru. Pronalaženje invarijanti im je obično bilo jednostavno, dok je najteži deo bilo definisanje postuslova problema, oko polovine vremena rešavanja problema je potrošeno na ovaj deo. U toku konstruisanja programa programeri su sami uočavali sitne greške koje pri uobičajenom programiranju često budu prevedene.

Na kursu održanom studentima doktorskih studija velika većina studenata je veoma lako i uspešno savladala koncepte invarijantnog programiranja dok im je najteži deo bio da nauče da koriste interaktivni dokazivač.

Studenti osnovnih studija su takođe uspešno savladali kurs i najteži deo im je predstavljalo dokazivanje uslova verifikacije i definisanje invarijanti za neke složene programe.

7 Zaključak

Invarijantno programiranje se pokazalo kao lako razumljivo i prihvatljivo od strane programera. U praksi postoje neki problemi koji invarijantnom programiranju otežavaju da bude opšte prihvaćeno i češće korišćeno. Jedan od razloga je što je potrebno predznanje iz logike koje veliki broj programera nema, drugo nema veliki broj alata koji bi odgovarali pri konstruisanju invarijantnih programa, i takođe možda najveća prepreka za korišćenje u industriji je što svaki program mora da se piše praktično kompletno jer još uvek ne postoje biblioteke kao kod uobičajenih programa koje danas pišemo. Ali ako bi se invarijantno programiranje više prihvatilo vremenom bi došli u situaciju da je razvijanje ovakvih programa jednako brzo kao i danas uobičajeno programiranje, a uz to dobijamo programe koji su korektni pri samoj konstrukciji.