Verifikacija softvera

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

Peleksic Ljubica, Milica Kojicic
peleksic.ljubica@gmal.com, milicakojicic@gmail.com

31. mart 2015.

Sažetak

Sadržaj

1 Uvod

2 Simboličko izvršavanje

U ovom odeljku ćemo se baviti simboličkim izvršavanjem programa. Radi se o tome da se umesto prosleđivanja normalnog ulaza programu, prosleđuju proizvoljne simboličke vrednosti. Izvršavanje programa se odvija normalno, osim što vrednosti mogu biti proizvoljne formule sastavljenje od ulaznih simbola. Tešski, ali i zanimljivi problemi koji nastaju pri simboličkom izvršavanju, potiču od uslovnih grananja. Proizvodnja velikih razmera pouzdanih programa je jedan od osnovnih uslova za primenu računara u izazovnim problemima današnjice. Neke tehnike se koriste u praksi, a neke su još uvek tema istraživanja. Program treba da ispunjava zahteve, iako nisu date formalne specifikacije. Neki uzorak podataka koji treba da bude obrađen od strane programa je dat programu. Ako je ocenjeno da program proizvodi korektan rezultat za dati uzorak podataka, pretpostavlja se da je on i tačan. Osnovno pitanje koje se postavlja je kako odabrati taj uzorak.

Skorašnji radovi o dokazivanju korektnosti programa formalnom analizom obecavaju mnogo i pokazuje se da je to ultimativna tehnika za pravljenje pouzdanih programa. Ali nije verovatno da će se svođenje teorije na praksu lako rešiti u skoroj budućnosti. Testiranje programa i dokazivanje korektnosti programa mogu se posmatrati kao ekstremne alternative. Dok testira, programer može biti siguran da za neki test primer u uzorku program radi tačno pažljivim proučavanjem rezultata. Tačnost izvršavanja za ulaze koji nisu u uzorku je i dalje podložan sumnji. S druge strane, pri dokazivanju korektnosti programa, programer formalno dokazuje da program pri svakom izvršavanju ispunjava zahteve koji su dati specifikacijom, a da pritom ne mora da izvršava program uopšte. Da bi to mogao, on daje preciznu specifikaciju tačnog ponašanja programa i dalje sledi formalnu proceduru dokazivanja da bi pokazao da su program i specifikacija konzistentni. Poverenje u ovaj metod zavisi od tačnosti pri kreiranju specifikacije i konstrukcije koraka u dokazu, kao i od mašinski zavisnih problema kao što su prekoračenje, zaokruživanje itd... Ovde se radi o praktičnom pristupu između ove dve krajnosti. Iz najprostijeg ugla, ovo je poboljšana vrsta tehnike testiranja. Umesto izvršavanja programa nad skupom ulaza, program se šimbolično"izvršava nad klasama ulaza. To znači da rezultat simboličkog izvršavanja može biti ekvivalentan velikom broju uobičajenih test primera. I ovi rezultati svakako mogu biti provereni od strane programera. Klasa ulaza, okarakterisana svakim simboličkim izvršavanjem, je određena u zavisnosti od kontrole toka programa nad ulazom. Ako je kontrola toka programa u potpunosti nezavisna od ulaznih varijabli, jedno simboličko izvršavanje će biti dovoljno da proveri sva moguća izvršavanja programa. Ako pak, kontrola toka zavisi od ulaza, mora se pribeći analizi slučajeva. Često, skup ulaznih klasa je koji pokrivaju sve moguće slučajeve je praktično beskonačan, tako da je ovo i dalje metoda testiranja. Kako god, ulazne klase su određene samo onim ulazima koji utiču na kontrolu toka i simboličko testiranje obećava postizanje boljih rezultata mnogo lakše od običnog testiranja.