UVOD

- formalne metode su matematicki zasnovane tehnike za specificiranje zahtjeva i arhitekture u oblikovanju i razvoju, te za verifikaciju sklopovskih i programskih sustava.

-doprinosi pouzdanosti i robusnosti konacnog proizvoda, smanjuje cijenu i skracuje time to market

- ne zamjenjuje testiranje vec su to dva medusobno komplementarna skupa tehnika

- staticko rasudivanje o sustavima, testiranje je dinamicko. Koriste alate za rasudivanje

- ASM i Z spadaju pod specifikaciju, SAT pod verifikaciju, SDL spec. i sinteza

-ASM je stroj s konacnim brojem generaliziranih stanja (konacan skup pravila)

- ASM = apstraktno stanje + apstraktni stroj

- rezultat je dokumentacija u ASM funkcijskom jeziku (npr. ASML)

- Z metoda ima state schemu (globalne izjave o sustavu), operation schemu (efekt odredenih operacija koje mijenjaju stanje sustava) i observation schemu (dohvat informacija, ne mijenjaju se podaci u sustavu)

- compiler : Analiza (leksicka, sintaksna, semanticka) i sinteza(generiranje prijelaznog koda, optimizacija koda, generiranje koda)

- Metode formalne verifikacije:

1) Provjera ekvivalentnosti (usporeduje novo oblikovani model i implementaciju s izvornim modelom i implementacijom)

2) Provjera modela (provjerava da li model implementacije zadovoljava zadano obiljezje)

3) Dokazivanje teorema (provjerava u nekom logickom formalizmu da li je implementacija istovjetna specifikaciji)

4) Provjera tvrdnje (provjerava se da neki specifican uvjet uvijek mora biti zadovoljen)

- FV metode mogu biti:

1) Vrlo lagane (neispravne, nekompletne, naivno pretrazivanje, efikasna i lagana tehnika za uporabu, dokazivanje obiljezja naivnim pretrazivanjem)

2) Srednje teske (Ispravne, nekompletne, analiza preko izracunavanja cvrste tocke)

3) Teske (ispravne i kompletne, verifikacija dokazivanjem logickih teorema, teske i zahtjevne za uporabu)

- Kombinacijska provjera - sustav se razbije na logicke konuse i usporeduju se izlazi za iste ulaze

- Sekvencijska provjera - kreira se zajednicki FSM od dva sustava i provjerava se ekvivalencija za svako valjano stanje sustava

- Provjera modela - automatizirana metoda provjere reaktivnih sustava modeliranih strojevima s konacnim brojem stanja na zadano obiljezje. Proces : implementaciju prikazi modelom, specifikaciju vremenskom logikom a verifikacija je automatizirana.

LOGIKA

- logike su formalni jezici koji predstavljaju informaciju na nacin da se mogu automatizirano izvoditi zakljucci

- sintaksa definira strukturu recenice u jeziku a semantika znacenje

- interpretacija je pridruzivanje true/false atomickim simbolima, evaluacija izrazu. Semantika ukljucuje interpretaciju i evaluaciju

- tautologija je uvijek istinit izraz (A v !A , istinita za svaku interpretaciju i evaluaciju), a kontradikcija uvijek neistinit (A & !A)

- dvije formule su semanticki ekvivalentne ako imaju istu bool vrijednost za svaku interpretaciju

- Eliminacija uvjeta: (A => B) = (!A v B)

- eliminacija dvostrukog uvjeta : (A <=> B) = ((A=>B) & (B=>A))

- modus ponens : P=T, (P=>Q) = T, generiraj Q = T

- modus tolens : !Q = T, (P=>Q) = T, generiraj !P

- L je konacan skup pravila zakljucivanja, `L je konacan skup ispravno definiranih formula (wff)

- sekvencija formula ili pojedina formula je teorem (dokaz,dedukcija) iz skupa formula `L ako je u skupu `L ili se moze izvesti iz `L

- skup `L je konzistentan akko ne sadrzi formule gdje bi w i !w istovremeno bili teoremi

(npr. {P,!Q,(P=>Q)} je nekonzistentan jer sadri !Q a preko Modus ponens se moze izvuc Q

- sustav je odrediv akko postoji algoritam koji hoce ili nece u konacnom vremenu odrediti teorem w

- sustav je poluodrediv ako ce u konacnom vremenu odrediti teorem ako postoji, inace nece (u konacnom vremenu odrediti da nije)

- interpretacija je model formalnog sustava ako evaluira sve njegove formule u istinito

- skup formula je zadovoljiv ako ima barem jedan model

- skup formula `L implicira formulu w ako je za svaki model od `L ujedno i model od w. Formula je tada logicka posljedica skupa formula `L (`L |= w)

- formalan sustav je ispravan ako svaka pravilima dokazana formula je ujedno i logicka posljedica skupa `L ( `L |- w implicira `L |= w)

- formalan sustav je kompletan ako je svaku logicku posljeidicu skupa `L moguce dokazati pravilima L ( `L |= w implicira `L |- w)

- u ispravnom i kompletnom formalnom sustavu vrijedi `L |= w = `L |- w, tj. logicka posljedica je ujedno i teorem

- propozicijska logika je ispravna, kompletna i odrediva, jer operira s konacnim skupom simbola

- ako zelis dokazati ekvivalentnost, dokazi da je ((alfa => beta) & (beta => alfa)) tautologija, odnosno da je njena negacija nezadovoljiva (dedukcija)

- DNF (k1 & k2 & k3) V (k4 & k5 & k6) -> da li je formula zadovoljiva

- CNF (k1 V k2 V...) & (k3 V k4 V...) -> da li je formula tautologija

- SAT problem (zadovoljivost) -> trazenje modela jedne slozene formule koja se sastoji iz konjukcije svih formula u `L ( `L je najcesce dan u CNF obliku)

- S je logicka posljedica `L ako je (`L & !S) nezadovoljiva

- predikatna logika je poluodrediva, ispravna i kompletna

- vremenska logika : odabiremo : propozicijska, globalna, grananje, buduce vrijeme

- beskonacno stablo izvodenja -> vremenska logika s grananjem -> CTL

-sigurnost (nesto lose se nece dogoditi), zivotnost (nesto dobro ce se konacno desiti)

- FV provjera modela : Sustav modeliran kao Kripke + CTL specifikacija idu na verifikaciju koja prolazi kroz sva stanja i javlja da li model implementacije logicki zadovoljava specifikaciju

LTL

- promatramo sustave koji se mogu modelirati strojevima s konacnim brojem stanja, reaktivne programe i analizira se ponasanje duz potencijalno beskonacnih putova izvodenja

- kripke se dekomponira u pojedinacne beskonacne sekvence

- ponasanje sustava je kolekcija beskonacnih sekvenci prijelaza

- LTL formula je istinita za neki sustav (Kripke strukturu) ako vrijedi za sve pojedinacne putove izvodenja tog sustava. Kvantifikator A je implicitan, ne postoji kvantifikator E

- jedna beskonacna sekvenca s pocetnim stanjem i oznacavanjem propozicijskih simbola koji vrijede u pojedinim stanjima je linearna vremenska struktura

- vremenska crta je totalno ureden skup stanja, linearna vremenska struktura je dana sa trojkom (S: konacan skup stanja, x: N->S beskonacna sekvenca stanja, L: S-> 2^AP oznacavanje stanja skupom propozicijskih simbola)

Usporedba LTL i CTL:

- Različita i neusporediva snaga izražavanja (ekspresivnost).

-CTL eksplicitno kvantificira putove pa bi se moglo reći da je izražajnija od LTL.

-S duge strane, LTL može selektirati sve pojedinačne putove iz nekog stanja koji zadovoljavaju LTL formulu pa bi se moglo reći da je izražajnija.

-Neke formule u CTL nije moguće izraziti u LTL-u i obratno.

-U LTL-u su složenije procedure provjere modela ali jednostavnije neke druge procedure (npr. valjanost).

-U LTL dozvoljena distribucija preko logičkih vezica i ugnježđivanje vremenskih operatora (u CTL ne)