Verifikacija softvera

— Apstraktna interpretacija —

Milena Vujošević Janičić

www.matf.bg.ac.rs/~milena

Matematički fakultet, Univerzitet u Beogradu

Pregled

Pregled

Apstraktna interpretacija (engl. *Abstract Interpretation*)

- Teorijski okvir za formalizaciju apstrakcije.
- Osnovna ideja: konkretna semantika programa je previše kompleksna da bi se o njoj moglo rezonovati. Zbog toga je potrebno apstrahovati konkretnu semantiku programa
- Uslov: ako se dokaže ispravnost u apstrahovanoj semantici, onda je potrebno osigurati da ta ispravnost važi i u konkretnoj semantici
- `http://www.di.ens.fr/~cousot/AI/IntroAbsInt.html`

Apstraktna interpretacija

Ideje

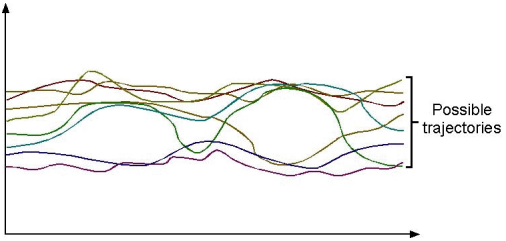
Osnovne ideje datiraju iz 1977. godine: *Patrick Cousot, Radhia Cousot: "Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints"*

Karakteristike

- Skalira dobro na velikim programima
- Ne propušta greške, ali može imati lažna upozorenja
- Primena u avio-industriji, automobliskoj industriji, svemirske letilice — neki standardi zahtevaju upotrebu statičke analize i apstraktne interpretacije.

Putanje kroz program

Konkretna semantika programa formalizuje skup svih mogućih izvršavanja programa u interakciji sa svim mogućim sredinama izvršavanja. Ako se izvršavanje predstavi kao kriva koja pokazuje promenu vektora x(t) vrednosti ulaza, stanja i izlaznih promenljivih programa kao funkciju vremena t, konkretna semantika onda može da se predstavi skupom krivih:



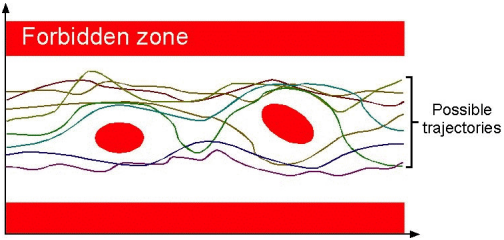
Neodlučivost

Konkretna semantika programa je jedan beskonačni matematički objekat koji nije izračunljiv: nije moguće napisati program koji može da predstavi i izračuna sva moguća izračunavanja za bilo koji program u svim njegovim mogućim uslovima izvršavanja.

Prema tome, sva netrivijalna pitanja o konkretnoj semantici programa su neodlučiva: nije moguće napisati program koji može da odgovori na bilo koje pitanje o izvršavanjima bilo kog programa (jer bi inače konkretna semantika programa morala da bude izračunljiva).

Zadavanje specifikacija

Sigurnosne svojstva programa izražavaju da ne postoji izvršavanje programa koje može da dođe do stanja greške. Grafički, skup ovih stanja može da predstavlja zabranjenu zonu (označenu crvenom bojom na slici).



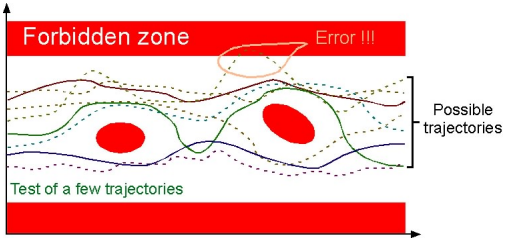
Dokaz ispravnosti

Dokazivanje sigurnosnih svojstava

Verifikacija sigurnosnog svojstva sastoji se u dokazivanju da je presek konkretne semantike programa sa zabranjenom zonom prazan. Kako je konkretna semantika neizračunljiva, problem verifikacije je neodlučiv. Nije uvek mougće da se odgovori na pitanje o sigurnosti programa kompletno automatski, sa konačnim resursima bez neizvesnosti u odgovor i bez ljudske intervencije.

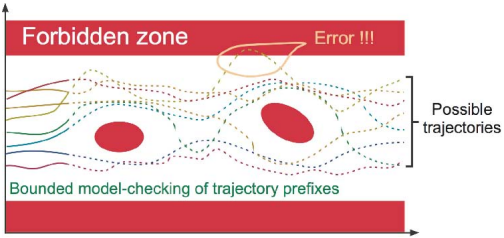
Testiranje i debagovanje

Testiranje/debagovanje se sastoji od razmatranja podskupa skupa svih mogućih izvršavanja:



Ograničeno proveravanje modela

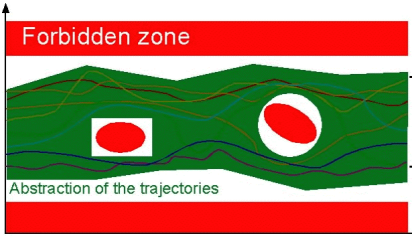
Ograničeno proveravanje modela se sastoji u istraživanju prefiksa svih mogućih izvršavanja:



Ograničeno proveravanje modela nije dokaz, jer greške koje se kasnije javljaju mogu da se promaše.

Apstraktna interpretacija

Apstraktna interpretacija se sastoji od razmatranja apstraktne semantike, koja je nadskup konkretne semantike programa:



Apstraktna semantika pokriva sve moguće slučajeve. Prema tome, ako je apstraktna semantika bezbedna, tj ne preseca se sa zabranjenom zonom, onda je takva i konkretna semantika.

Apstraktna interpretacija

- Semantika programa može se opisati konkretnim domenom D_c i relacijama nad ovim domenom. Ove relacije se mogu menjati tokom izvršavanja naredbi programa.
- Za velike programe, ispitivanje da li neko svojstvo važi nad domenom D_c otežano je zbog veličine samog domena, zbog velikog broja mogućih putanji kroz program kao i zbog neodlučivosti koja se javlja u raznim kontekstima.
- Jedan način prevazilaženja ovih poteškoća je aproksimacija konkretnog domena D_c apstraktnim domenom D_a, odnosno konkretan domen vrednosti zamenjuje se apstraktnim domenom opisa ovih vrednosti.

Apstraktna interpretacija - primer

- Iako apstraktni domen nije tako precizan kao konkretan domen, on može u nekim situacijama da dâ odgovore o važenju nekih svojstva.
- Da li je broj paran ili neparan?
- Konkretna interpretacija: proverimo ostatak pri deljenju sa dva
- Apstraktna interpretacija: proverimo poslednju cifru broja, tj konkretan domen: prirodni brojevi, apstraktni domen: cifre od 0 do 9, svaki broj se preslikava u svoju poslednju cifru
- Razmatranje poslednje cifre je, u opštem slučaju, efikasnije nego deljenje

Apstraktna interpretacija - primer

- Beskonačni domen skupa celih brojeva može se zameniti apstraktim domenom koji sadrži vrednosti znakova brojeva, tj. skupom $\{+, -, 0\}$.

$$\begin{aligned} a_0 &= \{0\} \\ a_+ &= \{n \mid n > 0\} \\ a_- &= \{n \mid n < 0\} \end{aligned}$$

Ovakva apstrakcija može da nam da potpuno precizan odgovor na pitanje znaka množenja dva broja:

$$\begin{aligned} 0 \times a_+ &= 0 \times a_- = a_+ \times 0 = a_- \times 0 = 0 \\ a_+ \times a_+ &= a_- \times a_- = a_+ \\ a_+ \times a_- &= a_- \times a_+ = a_- \end{aligned}$$

Apstraktna interpretacija - primer

- S druge strane, ova apstrakcija ne može da nam da precizan odgovor u kontekstu znaka sabiranja i oduzimanja dva broja:

$$\begin{aligned} a_+ + a_+ = a_+ - a_- = a_+ + 0 = 0 + a_+ = 0 - a_- &= a_+ \\ a_- + a_- = a_- - a_+ = a_- + 0 = 0 + a_- = 0 - a_+ &= a_- \\ a_+ + a_- = a_- + a_+ = a_+ - a_+ = a_- - a_- &= ? \end{aligned}$$

- Zato je potrebno proširiti skup apstrakcija tako da obuhvata i sve moguće brojeve.

$$\begin{aligned} a_0 &= \{0\} \\ a_+ &= \{n \mid n > 0\} \\ a_- &= \{n \mid n < 0\} \\ a &= \{n\} \end{aligned}$$

Apstraktna interpretacija - primer

- Sada je:

$$\begin{aligned}
0+0=0-0 &= 0 \\
a_+ + a_+ = a_+ - a_- = a_+ + 0 = 0 + a_+ = 0 - a_- &= a_+ \\
a_- + a_- = a_- - a_+ = a_- + 0 = 0 + a_- = 0 - a_+ &= a_- \\
a_+ + a_- = a_- + a_+ = a_+ - a_+ = a_- - a_- &= a \\
a + a_+ = a_+ + a = a_- + a = a + a_- &= a \\
a - a_+ = a_+ - a = a_- - a = a - a_- &= a \\
a + 0 = 0 + a = 0 - a = a - 0 &= a
\end{aligned}$$

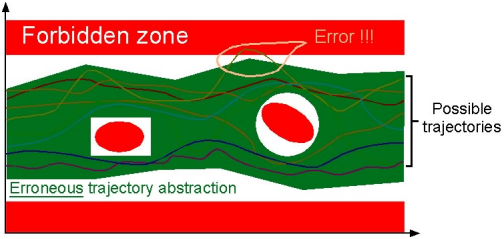
a nam je zapravo gubitak informacije, situacija u kojoj ne znamo ništa o znaku rezultata.

Izbor apstrakcije

- Apstrakcije se biraju u skladu sa problemom koji se ispituje
- Pravi izbor apstrakcije je suštinski za apstraktnu interpretaciju
- Primeri nekih apstraktnih domena koji se često upotrebljavaju `http://www.dsi.unive.it/~avp/domains.pdf`
- **Apstraktna semantika treba da bude saglasna, da ostane dovoljno precizna da se izbegnu lažna upozorenja, kao i da ostane dovoljno jednostavna da se izbegne fenomen kombinatorne eksplozije.**

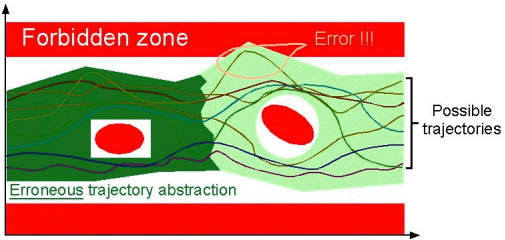
Pogrešne apstrakcije

U okviru formalnih metoda, apstraktne semantike moraju da se izaberu kao nadskup svih mogućih konkretnih semantika jer u suprotnom apstraktno reznovanje može da bude nekorektno za konkretno rezonovanje:



Pogrešne apstrakcije

Tehnike kao što su ograničeno proveravanje modela, iako formalne, ne istražuju sve moguće putanje već samo prefikse putanja, ili, u slučaju *refinement model-checking* mogu da se ne završe, što vode do toga da neke greške mogu da se promaše, odnosno da su ove tehnike nekorektne/nesaglasne.



Zaravo, ovo su alati za pronalaženje grešaka, ne za dokazivanje korektnosti.

Lažna upozorenja

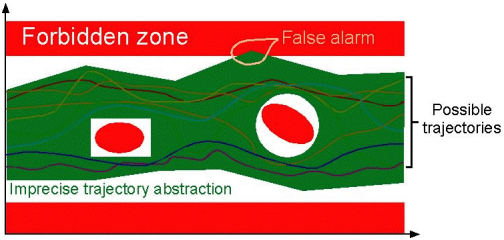
Apstraktne semantike na koju se formalne metode oslanjaju su:

- korektne/saglasne, odnosno su nadskup konkretne semantike i
- jednostavne, ili barem dovoljno jednostavne da mogu da se predstave u okviru mašine.

U odsustvu upozorenja, ovo vodi dokazu korektnosti programa. Ipak, posledica aproksimacije svih mogućih izvršavanja je da se razmatraju i neka nepostojeća izvršavanja, od kojih neka mogu da vode do greške, što onda vodi do lažnih upozorenja.

Lažna upozorenja

Lažna upozorenja odgovaraju slučajevima kada apstraktna semantika preseca zabranjenu zonu dok je konkretna semantika ne preseca. Dakle, signalizira se greška do koje ne može stvarno da dođe u realnosti:



Apstraktna interpretacija

- Apstrahovanjem se mogu izgubiti važne informacije — uzrok lažnih upozorenja
- Biranje adekvatne aproksimacije domena oslanja se na monotone funkcije u okviru parcijalno uređenih skupova, posebno na teoriju mreža (eng. *lattice theory*), na računanje fiksnih tačaka parcijalno uređenih skupova, na Galoaove veze (eng. *Galois connections*).
- Apstraktna interpretacija ima niz primena: kompilacija, određivanje invarijanti u verifikaciji, verifikacija: automobilska/avio/svemirska industrija.

Apstraktna interpretacija

Alati (sa skupim licencama)

- Astrée — AbsInt
- Polyspace Bug Finder — MathWorks
- Coverity — Synopsys

Alati

- CPAchecker — Free software, Apache 2.0 License
- Frama-C value analysis — Open source software

Literatura

http://www.di.ens.fr/~cousot/AI/

Na srpskom:

- Seminarski rad, Milan Čugurović: Intuicija.
- Seminarski rad, Dimitrije Špadijer: Teorija.