Formalizacija i automatizacija euklidske geometrije

Vesna Pavlović, Sana Stojanović
Matematički fakultet, Beograd
ARGO seminar, 04.06.2008.

Plan

- Formalno dokazivanje teorema
- Formalizacija euklidske geometrije
- Automatizacija euklidske geometrije

Plan

- Formalno dokazivanje teorema
- Formalizacija euklidske geometrije
- Automatizacija euklidske geometrije

Šta je dokaz? Šta je matematički dokaz?

Dokazati znači utvrditi postojanje, istinitost ili valjanost dokazom ili logikom. (Merriam-Webster)

U matematici, dokaz je demonstracija da, za dati skup aksioma, neka tvrdjenja koja su nam od interesa su nužno tačna. (Wikipedia)

Primer: $\sqrt{2}$ nije racionalan broj.

Dokaz: Pretpostavimo da postoji $r \in Q$ tako da važi $r^2 = 2$. Stoga postoje uzajamno prosti brojevi p i q tako da je $r = \frac{p}{q}$. Stoga je $2q^2 = p^2$, tj. p^2 je deljivo sa 2. 2 je prost broj, pa takodje deli i p, tj. $p = 2s$. Zamenom u jednačinu $2q^2 = p^2$ i deljenjem sa 2 dobijamo $q^2 = 2s^2$. Stoga, q je takodje deljivo sa 2. Kontradikcija.

Lepo, ali...

- I dalje, po nekima, nije dovoljno strogo.
  - Šta su aksiome? Šta su pravila?
  - Koliko "krupni" mogu biti koraci?
  - Šta je očigledno ili trivijalno?
- Neformalni jezik.

Šta je formalan dokaz?

- Izvodjenje u formalnom računu

$\Lambda_1, \Lambda_1, \ldots \Lambda_k$ - aksiome i prethodno dokazane teoreme

Formalni dokaz rečenice P je niz tvrdjenja

$$S_1, S_2, \ldots S_n$$

za koji važi:

1. S_n je P i važi jedno od sledećih:
   - S_i je jedno od $\Lambda_1, \Lambda_1, \ldots \Lambda_k$
   - S_i sledi iz prethodnih tvrdjenja na osnovu valjanog argumenta koristeći pravila zaključivanja

Primer formalnog dokaza

Primer: $A \wedge B \rightarrow B \wedge A$ se može izvesti u sledećem sistemu:

$\frac{X \in S}{S \vdash X}$ (pretpostavka) $\qquad$ $\frac{S \cup \{X\} \vdash Y}{S \vdash X \rightarrow Y}$ (impI)

$\frac{S \vdash X \quad S \vdash Y}{S \vdash X \wedge Y}$ (conjI) $\qquad$ $\frac{S \cup \{X, Y\} \vdash Z}{S \cup \{X \wedge Y\} \vdash Z}$ (conjE)

Dokaz:

1. $\{A, B\} \vdash B$ (pretpostavka)
2. $\{A, B\} \vdash A$ (pretpostavka)
3. $\{A, B\} \vdash B \wedge A$ (prema conjI iz 1 i 2)
4. $\{A \wedge B\} \vdash B \wedge A$ (prema conjE iz 3)
5. $\{\} \vdash A \wedge B \rightarrow B \wedge A$ (prema impI iz 4)

Značaj dobijanja formalnih dokaza

- Knjige i časopisi su puni pogrešnih dokaza
  (ne neophodno i pogrešnih tvrdjenja)

- Ispravnost softverskih i hardverskih komponenti
  mora biti potvrdjena što formalnije

Šta je formalna verifikacija?

Formalna verifikacija je procedura utvrdjivanja ispravnosti ili neispravnosti datog algoritma u odnosu na odredjenu formalnu specifikaciju ili osobinu, korišćenjem formalnih metoda matematike.

Dva pristupa u formalnoj verifikaciji:

1. Provera modela
   - sistematsko iscrpljujuće istraživanje matematičkog modela

2. Logičko zaključivanje
   - korišćenje formalne verzije matematičkog rezonovanja o sistemu

Šta je dokazivač teorema?

Implementacija formalne logike na računaru

- Potpuno automatizovan (iskazna logika)
- Automatizovan, ali se nužno ne zaustavlja (logika prvog reda)
- Sa automatizacijom, ali uglavnom interaktivan (logike višeg reda)
- Zasnovan na pravilima i aksiomama
- Mogu da daju formalne dokaze

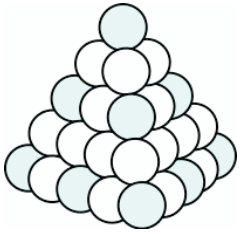
Dokazivači teorema

Najvažniji dokazivači teorema:

- HOL Light (John Harrison)
- Isabelle/Isar (Lawrence C. Paulson, Tobias Nipkow, Markus Wenzel)
- Coq (Benjamin Werner, Georgies Gonthier)
- Mizar (Andrzej Trybulec)
- ProofPower (Roger Jones, Rob Arthan)

Primer: Keplerova pretpostavka

Značaj formalizacije tvrdjenja čiji dokazi nisu tako očigledni i trivijalno razumljivi



- Popunjavanje velikog kontejnera malim sferama iste veličine
- Cilj je maksimizacija gustine uredjenja
  - Random packing - 65%
  - Cubic close packing $\frac{\pi}{\sqrt{18}} \simeq 0.74048.$
- Thomas Hales - dokaz iscrpljivanjem
  njegova formalizacija je ocenjena na 20 čovek - godina

"Sto najvećih teorema"

Paul & Jack Abad, 1999.

Kriterijumi:

- mesto koje teorema zauzima u literaturi
- kvalitet dokaza
- neočekivanost rezultata

Njihova formulacija i formalizacija mogu se naći na:
http://www.cs.ru.nl/~freek/100/

Oko 80% ovih teorema je formalizovano

Isabelle - Osnovni koncepti

- Interaktivno okruženje za dokazivanje teorema
- Naslednik HOL dokazivača teorema
- Prirodna dedukcija je glavni sistem izvodjenja
- Uključuje mehanizam za prezapisivanje termova i dokazivač koji radi po principu tabloa
- Koristi se za utvrdjivanje korektnosti sigurnosnih protokola, osobina semantike programskih jezika, formalizaciju teorema iz matematike i računarstva

Isabelle - Osnovni koncepti

Primer:

- Matematika: ako je $x < 0$ i $y < 0$ onda važi: $x + y < 0$
- Formalna logika: $\vdash x < 0 \wedge y < 0 \rightarrow x + y < 0$
  varijacija: $\{x < 0; y < 0\} \vdash x + y < 0$
- Isabelle: lemma " $x < 0 \wedge y < 0 \rightarrow x + y < 0$"
  varijacija: lemma: "$[|x < 0; y < 0|] \Rightarrow x + y < 0$"

- Isabelle/Isar: lemma
  assumes "$x < 0$" and "$y < 0$"
  shows "$x + y < 0$"

Plan

- Formalno dokazivanje teorema
- Formalizacija euklidske geometrije
- Automatizacija euklidske geometrije

Formalizacija Hilbertovog aksiomatskog sistema

- Euklidovi "Elementi"
- Hilbertove "Osnove geometrije"
- Formalizacija:
  - Christophe Dehlinger, Jean-Francois Dufourd, Pascal Schreck: Coq proof assistant
  - Jacques Fleuriot, Laura Meikle: Isabelle/Isar proof assistant

Formalizacija aksiomatskog sistema Tarskog

- Aksiomatski sistem Alfreda Tarskog

- Wolfram Schwabhauser:
  Metamathematische Methoden in der Geometrie

- Formalizacija:

  – Julien Narboux:
    Coq proof assistant

Formalizacija geometrije u proof assistant-u (Narboux)

- Prednosti:
  - daje visok stepen pouzdanosti dokaza koji su generisani
  - dozvoljava umetanje čisto geometrijskih argumenata unutar drugih vrsta dokaza
- Problem sa degenerisanim slučajevima
  - Izvor: aksiomatski sistem

Zašto aksiome Tarskog? (Narboux)

Prednosti:

- Jednostavne su (11 aksioma i 2 predikata)
- Dobre meta-matematičke osobine obezbedjuju vrlo visok nivo pouzdanosti u dokaze koji su generisani
- Generalizacija u druge dimenzije je laka

Mane:

- Raspored korišćenja lema je mnogo komplikovaniji
- Nije dobro podešen za učenje

Naš cilj

- Ideja formalizacije geometrijskog rezonovanja je prilično nova
- Umesto dokazivanja Hibertovih teorema "ručno" u Isabelle/CoQ imamo ideju automatizacije celokupnog procesa
- Dobijamo formalnu verifikaciju dokaza

Plan

- Formalno dokazivanje teorema
- Formalizacija geometrije
- Automatizacija euklidske geometrije

EUKLID - metod za automatsko dokazivanje teorema

- Autori: Predrag Janičić i Stevan Kordić
- Dokazuje geometrijske teoreme na intuitivan, geometrijski način
- Dokazi su predstavljeni u formi prirodnog jezika
- Nova forma zasnivanja geometrije i nova klasifikacija geometrijskih aksioma

Aksiomatika euklidske geometrije u sistemu EUKLID

- Osnovni simboli: logički simboli, promenljive, konstante
- Predikati (primitivni i definisani):

Predikat	Čitamo
$\mathcal{S}(a)$	a je tačka
$\mathcal{L}(b)$	b je prava
$\mathcal{P}(c)$	c je ravan
$a = b$	a je identično sa b
$\mathcal{I}(a, b)$	a je incidentno b
$\mathcal{B}(a, b, c)$	b se nalazi između a i c
$\mathcal{C}(a, b, c, d)$	(a, b) se poklapa (c, d)
$(a, b) \cong (c, d)$	(a, b) se poklapa (c, d)
$colin(a, b, c)$	a, b i c su kolinearne
$copl(a, b, c, d)$	a, b, c i d su koplanarne
$intersect(a, b)$	a i b se seku

Aksiomatika euklidske geometrije u sistemu EUKLID

- Ostali tipovi geometrijskih objekata (duž, trougao, krug, unutrašnjost kruga, itd.) mogu biti uvedeni pomoću definicija
- To bi dovelo do uvećanog segmenta "tradicionalne" geometrije koja može biti pokrivena ovom teorijom
- Razlozi za ovo:
  - Na ovaj način, bez korišćenja teorije skupova, možemo pokriti veliki deo uobičajenih geometrijskih komponenti
  - Možemo obezbediti koja svojstva želimo određeni objekti da imaju
  - Aksiomatski sistem izgrađen na ovaj način bi jos uvek očuvao nezavisnost aksioma

Tipovi aksioma

- Aksiome zasnovanosti (uvođenje tipova geometrijskih objekata i opsega osnovnih relacija)
- Aksiome jednakosti
- Neproduktivne aksiome
- Granajuće aksiome
- Produktivne aksiome
- Jako produktivne aksiome

Aksiome zasnovanosti

$$\forall x((\mathcal{S}(x) \wedge \neg\mathcal{L}(x) \wedge \neg\mathcal{P}(x)) \vee (\neg\mathcal{S}(x) \wedge \mathcal{L}(x) \wedge \neg\mathcal{P}(x)) \vee (\neg\mathcal{S}(x) \wedge \neg\mathcal{L}(x) \wedge \mathcal{P}(x)))$$

$$\forall x \forall y(\neg(\mathcal{S}(x) \wedge \mathcal{S}(y)) \wedge \neg(\mathcal{L}(x) \wedge \mathcal{L}(y)) \wedge \neg(\mathcal{P}(x) \wedge \mathcal{P}(y)) \Rightarrow \neg(x = y))$$

$$\forall x \forall y(\neg(\mathcal{S}(x) \wedge \mathcal{L}(y)) \wedge \neg(\mathcal{S}(x) \wedge \mathcal{P}(y)) \wedge \neg(\mathcal{L}(x) \wedge \mathcal{P}(y)) \Rightarrow \neg\mathcal{I}(x, y))$$

$$\forall x \forall y \forall z(\neg\mathcal{S}(x) \vee \neg\mathcal{S}(y) \vee \neg\mathcal{S}(z) \Rightarrow \neg\mathcal{B}(x, y, z))$$

$$\forall x \forall y \forall z \forall u(\neg\mathcal{S}(x) \vee \neg\mathcal{S}(y) \vee \neg\mathcal{S}(z) \vee \neg\mathcal{S}(u) \Rightarrow \neg(x, y) \cong (z, u))$$

Aksiome jednakosti i saglasnosti

$$\forall x\,(x = x)$$

$$\forall x\,\forall y\,(x = y \Rightarrow y = x)$$

$$\forall x_1 \forall x_2 \ldots \forall x_n \forall y\ (x_i = y \wedge \Phi(x_1, x_2, \ldots, x_i, \ldots, x_n)$$

$$\Rightarrow \Phi(x_1, x_2, \ldots, y, \ldots, x_n))$$

Neproduktivne i granajuće aksiome

1. **Neproduktivne**

   Ako je tačka A incidentna pravoj p, i prava p incidentna ravni ϕ, onda je tačka A incidentna ravni ϕ

2. **Granajuće**

   Za svake tri tačke koje se nalaze na jednoj pravi uvek postoji tačno jedna od njih koja se nalazi između druge dve.

Produktivne i jako produktivne aksiome

1. **Produktivne**

   Ako za dve različite ravni postoji jedna tačka koja im je incidentna onda postoji najmanje još jedna tačka koja im je incidentna

2. **Jako produktivne**

   Postoje četiri različite nekoplanarne tačke

Klasifikacija i struktura aksioma

Sve aksiome (osnovne i izvedene), teoreme i definicije, pokrivene EUKLID-om imaju jednu od sledećih formi:

$$\forall x_1 \forall x_2 \ldots \forall x_n \exists y_1 \exists y_2 \ldots \exists y_m (\Phi(x_1, x_2, ..., x_n) \Rightarrow \quad (1)$$

$$\Psi(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m)) \quad (n, m \geq 1)$$

$$\forall x_1 \forall x_2 \ldots \forall x_n \exists y_1 \exists y_2 \ldots \exists y_m (\Phi(x_1, x_2, ..., x_n) \Rightarrow \quad (2)$$

$$\Psi_1(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m) \vee \Psi_2(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m) \vee ...$$

$$... \vee \Psi_k(x_1, x_2, ...x_n, y_1, y_2, ...y_m)) \quad (n, m \geq 1)$$

Klasifikacija i struktura aksioma

$$\forall x_1 \forall x_2 \ldots \forall x_n (\Phi(x_1, x_2, ..., x_n) \Rightarrow \Psi(x_1, x_2, ..., x_n)) \quad (n \geq 1) \quad (3)$$

$$\exists y_1 \exists y_2 \ldots \exists y_m (\Psi(y_1, y_2, ..., y_m)) \quad (m \geq 1) \quad (4)$$

gde su Φ, Ψ i Ψ_i literali nad nekim od promenljivih $x_1, x_2, ..., x_n$, odnosno $y_1, y_2, ..., y_m$

Algoritam izvođenja dokaza u dokazivaču EUKLID

- Aksiomatski sistem je baza algoritma
- Algoritam je nezavisan od konkretne kompjuterske implementacije
- Algoritam pokriva jednu klasu teorema
- Generisani dokazi odgovaraju tradicionalnim dokazima
- U osnovnoj verziji, dedukcija dokaza je usmerena klasifikacijom aksioma i njihovim redosledom unutar grupa
- Puno prostora za heuristike

Algoritam izvođenja dokaza u dokazivaču EUKLID

- *Dopustivi objekti* - pojam izveden iz principa *graničnika* koji se koristi u dokazivaču.

- Na početku izvođenja dokaza, skup dopustivih objekata je prazan i tokom izvođenja dokaza taj skup se proširuje pod kontrolom graničnika koji onemogućava pojavljivanje "beskonačnih grana" u dokazu

- Tokom izvođenja dokaza proširuje se, sukcesivnom primenom aksioma, *prostor znanja* koji sadrži činjenice o objektima čija je egzistencija utvrđena

Algoritam izvođenja dokaza u dokazivaču EUKLID

- Algoritam je parcijalno zasnovan na "metodi iscrpljivanja". Da bi se maksimalno ograničilo uvođenje novih geometrijskih objekata i činjenica nepotrebnih za dokaz i da bi se povećala efikasnost, aksiome su podeljene u grupe i poređane u okviru svake od njih

- Dobijeni dokazi mogu biti (automatski) optimizovani, tako da ne sadrže nepotrebne korake

- Metod korišćen u EUKLID-u je "forward chaining" i povezan je sa Herbrand-ovom teoremom. Nije mnogo efikasan ali omogućava automatsko izvođenje velikog broja dokaza koji se trenutno izvode ručno

Program EUKLID

- PROLOG verzija
- C verzija
- C++ verzija

Program EUKLID - C verzija

- Baza znanja je predstavljena kao skup nizova (po jedan niz za pozitivnu i jedan za negativnu formu predikata).

- Relacije ne sadrže informaciju o tipu objekta.

  Na primer: $I(1,2)$ i $S(1)$ i $L(2)$

- Jedan brojač za sve elementarne objekte (tačke, prave i ravni).

- Za svaki niz, indeks poslednje dodate činjenice se čuva (LIFO lista).

Program EUKLID - C verzija

- Statička organizacija podataka (prednost nad dinamičkom zbog brzine jednostavnih operacija: dodavanje novih činjenica i brisanje poslednje dodate).

- Aksiome su predstavljene pomoću funkcija (unutar kojih se dodaju nove činjenice u bazu znanja i izlaz koraka izvođenja u formi prirodnog jezika).

- Aksiome su hard-kodirane. Ne možemo dodavati nove leme i teoreme u sistem.

Jedna od aksioma u C verziji dokazivača EUKLID

- If a point A lies on a line p, and line p lies on a plane α than point A lies on a plane α

- ```
  int ax_n10() { int i1,i2; int x,y,z;

  for(i1=1;i1<=INC[0].index;i1++) {
    x=INC[i1].arg1; y=INC[i1].arg2;
    for(i2=1;i2<=INC[0].index;i2++)
      if (INC[i2].arg1==y)
      {
        z=INC[i2].arg2;
        if (!inc(x,z))
        {
          add_inc(x,z);
  ```
  ```

```
        sprintf(OUT,"Ako tacka %i pripada pravoj %i i ",x,y);
        sprintf(OUT,"prava %i pripada ravni %i, ",y,z);
        sprintf(OUT,"onda tacka %i pripada ravni %i",x,z);
        output(DEPTH,OUT);
        return 1;
      }
    }
} return 0; }
```

Format ulaznog fajla

Ako su p i q dve različite prave koje se seku onda postoji ravan koja ih sadrži:

```
premise line(1) line(2) not_identical(1,2) intersect(1,2)

theorem plane(-1) incident(1, -1) incident(2, -1)
```

Program EUKLID - C++ verzija

- Baza znanja:
  - Tačke, prave i ravni su predstavljene različitim brojačima.
  - Relacije sadrže informacije o tipovima objekata na koje se odnose. Višestruke relacije incidencije, jednakosti, itd.

    Primer: $I_1(1,2)$ znači da je 1 tačka, a 2 prava
- Sada ne moramo da proveravamo da li je objekat u bazi, dovoljno je da proverimo da li je njegov broj manji ili jednak od brojača za taj tip objekta.
- Svojstva objekata se ne čuvaju na isti način kao ranije. Čuvamo samo hash-eve svojstava (s obzirom da su nam jedine dve operacije provera svojstva i ubacivanje novog svojstva).

Program EUKLID - C++ verzija

- Aksiome i teoreme su interpretirane na isti način:

```
class Statement{
        vector<Property> _from;    // premise
        vector<Property> _have;    // zakljucak
        Statement* _by;            // tvrdjenje koje je primenjeno
}
```

- Jednom dokazano tvrđenje možemo ubaciti u bazu znanja i koristiti ga kao aksiomu.

Program EUKLID - C++ verzija

- Ideje:
  - Tačke, prave i ravni mogu biti prestavljene kao relacije takođe
  - Relacijama može biti dozvoljeno da unifikuju objekte (iz razloga što su u njima sadržane sve informacije potrebne za unifikaciju)
  - Promena redosleda premisa u teoremama

    Primer:

    Umesto uobičajenog redosleda premisa:

    `L(1) || S(1) S(2) Diff(1,2) I1(1,1) I1(2,1)`

    možemo prerasporediti premise tako da se relacije nalaze neposredno nakon elementarnih objekata koje opisuju:

    `L(1) || S(1) I1(1,1) S(2) Diff(1,2) I1(2,1)`

Program EUKLID - C++ verzija

Izlaz iz programa:

- Prirodni jezik
- Isabelle/Isar
- Coq