WAI 3720
Loogika arvutiteaduses

Tarmo Uustalu

TTÜ arvutiteaduse instituut

tarmo@cs.ioc.ee

Sügis 2002

KURSUSE PRAKTILINE KORRALDUS

- Tunniplaan:
  kolmap 11.00-13.30 loeng 3 h, esmasp (paaris nädalad) 10.00-11.30 harjutus 2 h.
- Teadmiste kontroll: Semestri jooksul 3–4 kiirkontrolltööd (ette teatatud kuupäevadel). Lõpus kirjalik eksam.
- Õppematerjal: Otsene õpik puudub, peamiselt õppejõu slaidid. Täiendavat lugemist olemas nii eesti kui ka inglise, vene k.
- Õppejõud: dots T Uustalu, `tarmo@cs.ioc.ee`, (0) 620 4250.
  Konsultatsioonid semestri vältel eelneval kokkuleppel emailitsi.
- Veebilk info ja materjalidega tekib aadressile `www.cs.ioc.ee/˜tarmo/lcs02/`.

- Eestikeelseid tekste:
  - P Lorents. Informaatika teoreetilised alused: struktuurne aspekt. EBS, 2001.
  - P Lorents. Keel ja loogika. EBS, 2000.
  - T Tamme, T Tammet, R Prank. Loogika: Mõtlemisest tõestamiseni. 2. trükk. TÜ, 2002.
  - P Lorents. Matemaatilise loogika põhimõisteid 1–22. Arvutustehnika & Andmetöötlus, 2(7)–4(7), 1988–1990.
  - R Prank. Matemaatiline loogika ja diskreetne matemaatika I, II, III. TRÜ, 1978, 1978, 1983.
  - (Oluliselt rohkem ei olegi, ainult veel mõned konspektid.)

WAI 3720 Loogika arvutiteaduses Sügis 2002 / Tarmo Uustalu Loeng 1

- Ingliskeelseid tekste:
  - M R A Huth, M D Ryan. Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge Univ Press, 2000.
  - D van Dalen. Logic and Structure. 3rd, augmented ed. Springer-Verlag, 1994.
  - R Lalement. Computation as Logic. Masson / Prentice Hall, 1993.
  - A Nerode, R A Shore. Logic for Applications. Springer-Verlag, 1993.
  - M Ben-Ari. Mathematical Logic for Computer Science. 2nd ed. Springer-Verlag, 2001.
  - Jne jne jne.

- Kursuse sisu:
  - Lauseloogika: süntaks, semantika, tautoloogiakontrolli meetoditest ja tõestussüsteemidest.
  - Predikaatloogikast.
  - Modaalsetest lauseloogikatest, sh temporaalloogikatest, dünaamilisest loogikast, teadmise/tõekspidamise loogikatest.
  - Intuitsionistlikust lauseloogikast.
  - Rakendustest eri liiki teadmiste esitamisel ja nende üle arutamisel (sh verifitseerimises programmide käitumiste kohta), programmikeeltes (loogikapõhiste programmeerimisparadigmade juures nagu loogiline programmeerimine, andmebaasid, funktsionaalprogrammeerimine).

LOOGIKA ARVUTITEADUSES

- Loogika (matemaatilise loogika mõistes) ning teoreetiline arvutiteadus on täna väga tihedalt läbi põimunud. Kumbki lahendab teise probleeme, püstitab teisele uusi probleeme.
- Loogika on täna igasuguse arvutiteaduse olulisim matemaatiline alusdistsipliin.
- Valdkonnad, mis loogikat enim kasutavad: programmikeelte tehnoloogia, tarkvaratehnoloogiad, intellektitehnika.
- Kiired arengud uurimistöös, aga ka rakendustes.
- Suurimad üldised arvutiteadusliku loogika konverentsid on LICS (logic in computer science), CSL (computer science logic), lisaks kümneid spetsiaalseid, nt CADE (conf on automated deduction), RTA (rewriting techniques and applications), TLCA (typed lambda calculi and applications).

- Illustreerimaks valdkonna ulatust:

  LICS 2003 scope

  Suggested, but not exclusive, topics of interest for submissions include: automata theory, automated deduction, categorical models and logics, concurrency and distributed computation, constraint programming, constructive mathematics, database theory, domain theory, finite model theory, formal aspects of program analysis, formal methods, hybrid systems, lambda and combinatory calculi, linear logic, logical aspects of computational complexity, logics in artificial intelligence, logics of programs, logic programming, modal and temporal logics, model checking, programming language semantics, reasoning about security, rewriting, specifications, type systems and type theory, and verification.

- Oluline rõhk kursuses:
  - Seosed programmeerimisega pole mitte üksnes tehnilised, vaid esimene algab juba mentaliteedi tasemelt:

    Nii nagu programmeerimine on keeltest, milles kirja panna ideid nõnda, et masingi neist aru saaks, ning kunstist, kuidas seda teha, samuti on ka loogika.
    $\rightarrow$ Süntaksi (lingvistilised objektid) ja semantika (objektid out there) selge eristamise vajalikkus, keele valiku määrav roll info esitamise mugavuse ja esituse kasutatavuse juures, keele semantika kommunikeerimise probleemid . . .
- Kursus on matemaatiline!

LAUSELOOGIKA: SÜNTAKS

- Lauseloogika (propositional logic) *signatuur* on mingi tähestik $\mathrm{PC} = \{p, q, \dots\}$, mille sümboleid nimetatakse lausesümboliteks (proposition symbols).
- Lauseloogilised *valemid* (formulae) (üle selle signatuuri) on hulk väljendeid ehk keel Fma, mis on defineeritud induktiivselt järgmiste tingimustega:
  - kõik lausesümbolid on valemid (nn atomaarvalemid, atomic formulae);
  - $\top$ (verum, tõde), $\bot$ (falsum, väärus) on valemid;
  - kui A on valem, siis $\neg A$ (mitte-A) on samuti valem;
  - kui A, B on valemid, siis $A \wedge B$ (A ja B), $A \vee B$ (A või B), $A \supset B$ (kui A, siis B e A implitseerib B) on ka valemid.
- Sümboleid $\top$, $\bot$, $\neg$ (eitus, negation), $\wedge$ (konjunktsioon), $\vee$ (disjunktsioon), $\supset$ (implikatsioon) kutsutakse loogilisteks konnektiivideks.

  ($\top$, $\bot$ on 0-kohalised, $\neg$ 1-kohaline, $\wedge$, $\vee$, $\supset$ 2-kohalised.)

- Näiteid:
  - $p \vee \neg p$, $p \supset q \wedge \neg r$ on valemid

    (kokkuleppeliselt seob $\supset$ nõrgemini kui $\wedge$, $\vee$ ja need omakorda nõrgemini kui $\neg$, st viimane valem on konkreetsüntaks valemile $p \supset (q \wedge \neg r)$);
  - kui A, B on valem ja p on lausesümbol, siis $A[B/p]$ (väljend, mis saadakse p kõigi A-s esinemiste (occurrences) asendamisel B-ga) on ka valem.
- Tähelepanek: Valemite hulk üle signatuuri on defineeritud induktiivselt.

  Järeldus: kui kõigil lausesümbolitel kui valemitel on mingi omadus P ning kui iga konnektiivi rakendamisel omadust P evivatele valemitele saadakse valem, millel taas on omadus P, siis on kõigil valemitel omadus P.

  Iga valem on kas atomaarne või mingi konnektiivi rakendus teistele valemitele (veel enam, selline analüüs on alati unikaalne).

  Ühtegi valemit ei saa dekomponeerida lõputult (tema moodustamispuu on fundeeritud).

- Valemi *alamvalem* (subformula) on temas alamväljendina esinev valem. (Kuidas defineerida see mõiste matemaatiliselt?)
- Näide: Valemi $p \supset q \vee (\neg q \wedge r)$ alamvalemid on tema ise, p, $q \vee (\neg q \wedge r)$, $\neg q \wedge r$, $\neg q$, q ja r, kusjuures kõik peale q esinevad 1 kord, q aga 2 korda.
- Igas valemis on nii palju alamvalemite esinemisi kui temas on lausesümbolite ja loogiliste konnektiivide esinemisi. (Miks?)

LAUSELOOGIKA: SEMANTIKA

- Olgu $\mathrm{PC} = \{p, q, \dots\}$ fikseeritud lauseloogiline signatuur, st lausesümbolite tähestik.
- *Interpretatsioon* on siis suvaline funktsioon $I : \mathrm{PC} \to \{1, 0\}$ ehk tõeväärtuse (truth value) 1 (tõene, true) või 0 (väär, false) omistus igale lausesümbolile.
- Lausearvutuse valemite *väärtustus* (valuation) interpretatsioonis I on funktsioon $[\![\cdot]\!]^I : \mathrm{Fma} \to \{1, 0\}$, mis on defineeritud induktsiooniga järgmiselt:
  - $[\![p]\!]^I = I(p)$, kui p on lausesümbol;
  - $[\![\top]\!]^I = 1$, $[\![\bot]\!]^I = 0$;
  - $[\![\neg A]\!]^I = 1 - [\![A]\!]^I$;
  - $[\![A \wedge B]\!]^I = \min([\![A]\!]^I, [\![B]\!]^I)$;
  - $[\![A \vee B]\!]^I = \max([\![A]\!]^I, [\![B]\!]^I)$;
  - $[\![A \supset B]\!]^I = \max(1 - [\![A]\!]^I, [\![B]\!]^I)$.

- Öeldakse, et I *kehtestab* (satisfies) A, A *kehtib* (holds) I-s, A on I-s tõene ehk I on A *mudel* (tähistus $I \models A$), kui $[\![A]\!]^I = 1$;

  I *väärab* (falsifies) A, A on I-s väär ehk I on A *kontramudel* (countermodel) (tähistus $I \not\models A$), kui $[\![A]\!]^I = 0$.

- Näeme, et iga I korral
  - $I \models p$ parajasti siis, kui $I(p) = 1$, kui p on lausesümbol;
  - $I \models \top$ alati; $I \models \bot$ mitte kunagi;
  - $I \models \neg A$ parajasti siis, kui $I \not\models A$;
  - $I \models A \wedge B$ parajasti siis, kui $I \models A$ ja $I \models B$;
  - $I \models A \vee B$ parajasti siis, kui $I \models A$ või $I \models B$;
  - $I \models A \supset B$ parajasti siis, kui $I \not\models A$ või $I \models B$.

- Öeldakse, et A on *üldkehtiv* (valid), *tautoloogiline* ehk *loogiliselt tõene* (tähistus $\models A$), kui A kehtib igas interpretatsioonis;

  A on *kehtestamatu* (unsatisfiable), *vastuoluline* (contradictory) ehk *loogiliselt väär*, kui ta ei kehti üheski interpretatsioonis.

  A on *kehtestatav* (satisfiable), kui A kehtib mõnes interpretatsioonis;

  A on *vääratav* (falsifiable) (tähistus $\not\models A$), kui ta mõnes interpretatsioonis ei kehti.

- A on tautoloogia parajasti siis, kui $\neg A$ on vastuolu; A on vastuolu parajasti siis, kui $\neg A$ on tautoloogia;

  A on kehtestatav parajasti siis, kui $\neg A$ on vääratav; A on vääratav parajasti siis, kui $\neg A$ on kehtestatav.

- Valemite hulga Γ kohta öeldakse, et ta *tingib* (entails) valemi B või et B on Γ *loogiline järeldus* (logical consequence) (tähistus $\Gamma \models B$), kui iga interpretatsiooni I korral, $I \models A$ ($A \in \Gamma$) implitseerib $I \models B$.
- $\{A_1, \ldots, A_n\} \models B$ parajasti siis, kui $\models A_1 \wedge \ldots \wedge A_n \supset B$.
- $\emptyset \models B$ parajasti siis, kui $\models B$ (ehk loogiline jrelduvus thjast valemite hulgast on sama, mis loogiline tesus).

WAI 3720 Loogika arvutiteaduses Sügis 2002 / Tarmo Uustalu Loeng 1

- Valemid A, B on *loogiliselt ekvivalentsed* (tähistus $A \Leftrightarrow B$), kui iga interpretatsiooni I korral $I \models A$ parajasti siis, kui $I \models B$.
- $A \Leftrightarrow B$ parajasti siis, kui $\models A \equiv B$ (ehk loogiline ekvivalents on sama, mis ekvivalentsi loogiline tõesus).

  [Siin $A \equiv B$ (A parajasti siis, kui B, A B-ga samaväärne) pole ametlik süntaks, vaid lühendab valemit $(A \supset B) \wedge (B \supset A)$.]
- Loogiline ekvivalents on ekvivalentsiseos: ta on refleksiivne ($A \Leftrightarrow A$), sümmeetriline (kui $A \Leftrightarrow B$, siis $B \Leftrightarrow A$) ning transitiivne (kui $A \Leftrightarrow B$ ja $B \Leftrightarrow C$, siis $A \Leftrightarrow C$).

- Olulisi tautoloogiaid lauseloogikas:

$$
\begin{gathered}
\top \\
\bot \supset A \\
A \vee \neg A \\
A \supset A \\
A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C) \\
A \equiv A \vee A \\
\neg(A \wedge B) \equiv \neg A \vee \neg B \\
A \equiv \neg\neg A \\
A \supset (B \supset A) \\
A \supset A \vee B \\
A \wedge B \supset C \equiv A \supset (B \supset C) \\
((A \supset B) \supset A) \supset A
\end{gathered}
$$

- Ekvivalentsete asendamise omadus: Kui $B \Leftrightarrow C$, siis $A[B/p] \Leftrightarrow A[C/p]$.
- Näide: Kuna $A \wedge B \Leftrightarrow B \wedge A$, siis $(A \wedge B) \wedge C \Leftrightarrow (B \wedge A) \wedge C$. Et pealegi $(B \wedge A) \wedge C \Leftrightarrow B \wedge (A \wedge C)$, siis transitiivsuse põhjal $(A \wedge B) \wedge C \Leftrightarrow B \wedge (A \wedge C)$.

- Probleemi teha kindlaks, kas etteantud valem on üldkehtiv või kehtestatav nimetatakse *üldkehtivus*- resp. *kehtestatavuskontrolliks* (validity checking, satisfiability checking) (tähistused TAUT, SAT).
- Meenutagem (mingist muust kursusest), et jah/ei probleemi nimetatakse *lahenduvaks*, kui leidub algoritm, mis iga sisendi korral peatub ja vastab korrektselt kas jah või ei.

  Teda nimetatakse *poollahenduvaks* (semidecidable), kui leidub algoritm, mis jah-vastust vääriva sisendi korral peatub ja vastab jah, ei-vastust vääriva sisendi korral peatub ja vastab ei või ei peatu.
- Lauseloogika puhul on lihtne näha, et TAUT ja SAT on lahenduvad. Valemi A väärtustust interpretatsioonis I mõjutavad I väärtused ainult nende lausesümbolitel, mis A-s vähemalt 1 kord esinevad. Neid ei saa olla rohkem kui sümbolite koguarv A-s. Järelikult piisab max 2^n juhu läbivaatamisest, kus $n = |A|$; seda kutsutakse tõeväärtustabelite meetodiks.

WAI 3720 Loogika arvutiteaduses Sügis 2002 / Tarmo Uustalu Loeng 1

- Näide: Teeme kindlaks, et $(p \wedge q \supset r) \supset (p \supset (q \supset r))$ on tautoloogia.

p	q	r	$p \wedge q$	$p \wedge q \supset r$	$q \supset r$	$p \supset (q \supset r)$	$(p \wedge q \supset r) \supset (p \supset (q \supset r))$
1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	1
1	0	1	0	1	1	1	1
1	0	0	0	1	1	1	1
0	1	1	0	1	1	1	1
0	1	0	0	1	0	1	1
0	0	1	0	1	1	1	1
0	0	0	0	1	1	1	1