

T-79.5101

Laskennallisen logiikan jatkokurssi (4 op) L

Kevät 2007

Kurssin sisältöä

- Modaalilogiikka
 - syntaksi
 - semantiikka
 - todistusmenetelmät
- Temporaalilogiikan sovellutukset rinnakkaisissa ja hajautetuissa järjestelmissä.
- Tietojenkäsittelyssä usein toistuvat käsitteet:
 - formaali malli
 - ristiriidattomuus ja täydellisyys
 - tehokkuus; ongelmien (laskennallinen) vaativuus

Käytännön asioita

Esitiedot: T-79.3001/144 Logiikka tietotekniikassa: perusteet tai vast.

Luennot: keskiviikkoisin klo 10–12, sali TB353

Luennoitsija: Prof. Ilkka Niemelä, TB337,
puh. 451 3290, e-mail: Ilkka.Niemela@tkk.fi.

Laskuharjoitukset: maanantaisin klo 12–13, sali TB353

Laskuharjoitusassistentti: DI Matti Järvisalo, TB354,
puh. 451 2896, e-mail: mjj@tcs.hut.fi.

Kotisivu: <http://www.tcs.hut.fi/Studies/T-79.5101/>

Uutisryhmä: opinnot.tik.logiikka

Sähköposti: t795101@tcs.hut.fi

Materiaali

- Luennot (katso kotisivu)
- M. Fitting: *Basic Modal Logic*, Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 1, Logical Foundations, 1993.
- E. Clarke, O. Grumberg and D. Peled: *Model Checking*, The MIT Press, 1999. (Chapters 1-4)
- E.A. Emerson: *Automated Temporal Reasoning about Reactive Systems*, Logics for Concurrency, F. Moller and G. Birtwistle (Eds.), Springer-Verlag, 1996, LNCS 1043, pp. 39–99.

Kurssin suorittaminen

- Kurssin suorittaminen edellyttää hyväksytyjä
 - kotilaskuja (3 tehtävää)
 - **tenttiä**
- Arvosana: tentin arvosana
- Opiskelija voi saada korkeintaan 4 ponustenttipistettä seuraavasti (tentin maksimipistemäärä 50):
 - Laskuharjoitusaktiivisuus (korkeintaan 3p.):
 - läsnä vähintään 4 kertaa / 1 p.
 - läsnä vähintään 8 kertaa / 2 p.
 - läsnä vähintään 12 kertaa / 3 p.
 - Palautelomakkeen täyttäminen (1 p.)

Modaalilogiikka tietojenkäsittelyssä

- Modaalilogiikka on käsitteiden *välttämätön, aikominen, tietäminen, uskominen, tuleva, mennyt, todistuva, tosi tapahtuman jälkeen, ...* logiikkaa
- Järjestelmien määrittely, analyysi, verifiointi: rinnakkaiset ja hajautetut järjestelmät, reaktiiviset järjestelmät, tietoliikenneprotokollat, tietoturva
- Tietämyksen esittäminen, luonnollisen kielen käsittely, ohjelmistoagentit, semanttinen web, ontologiat, ...

Logiikka tietojenkäsittelyssä

- Formaalit menetelmät yleistyvät
- Logiikkapohjaisten työkalujen käyttö tavallista
 - The use of formal verification tools is well established and becoming more so. Simulation- and emulation- based methodologies aren't sufficient to guarantee correctness with today's complex chips.*
- (Carl Pixley, Motorola Inc. in IEEE Spectrum, Jan 1997, p. 61)
- Tähän kehitykseen on vaikuttanut
 - Tietokoneiden suorituskyvyn ja muistin määrän kasvu
 - Toteutusmenetelmien kehittyminen

Järjestelmäsuunnittelun haasteita

- Järjestelmien määrittely ja suunnittelu virhealttiita
- Suunnittelun alkupään virheet kalliita korjata myöhemmin
- Tarvitaan matemaattisia menetelmiä



Rinnakkaiset ja hajautetut järjestelmät

- Useita rinnakkaisia ja hajautettuja prosesseja
- Jaetut resurssit, koordinointi, kommunikointi
- Toimivat keskeytyksettä
- Esimerkkejä: käyttöjärjestelmät, tietoliikenneprotokollat, laitteistokomponentit, ohjausjärjestelmät, ...
- Tällaisten järjestelmien suunnittelu on haastavaa ja tähän tarvitaan uudenlaisia menetelmiä

Esimerkki. (jatkoa)

Temporaalilogiikka soveltuu tähän tarkoitukseen:

- P*: aina (tulevaisuudessa) *P* totta.
- ◇*P*: joskus (tulevaisuudessa) *P* totta.

- (i) □◇*ex*
- (ii) □*en* → □◇*ex*
- (iii) □◇*en* → □◇*ex*

Erilaisia sovellusmahdollisuuksia:

1. Mallintarkastus: Täyttääkö annettu malli halutut ominaisuudet?
2. Toteutuvuus: Onko annetut ehdot täyttäviä malleja?
3. Pätevyys: Onko kaikilla annetut ehdot täyttävillä malleilla halutut ominaisuudet?

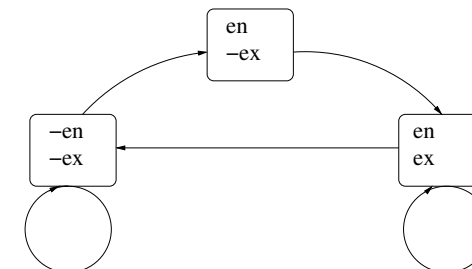
Esimerkki. (Reiluus)

Miten ilmaistaan täsmällisesti ja verifioidaan esimerkiksi seuraaventyypisiä ehtoja?

- (i) Jokainen prosessi pääsee suoritettavaksi äärettömän usein.
- (ii) Jokainen prosessi, joka on aina ajovalmis, pääsee suoritettavaksi äärettömän usein.
- (iii) Jokainen prosessi, joka on äärettömän usein ajovalmis, pääsee suoritettavaksi äärettömän usein.

☞ Tarvitaan **malli** järjestelmän käyttäytymiselle ja **kieli**, jolla voidaan ilmaista mallin ominaisuuksia.

Esimerkki. (jatkoa)



- (i) □◇*ex* ?
- (ii) □*en* → □◇*ex* ?
- (iii) □◇*en* → □◇*ex* ?