T-79.5102

Laskennallisen logiikan erikoiskurssi (4 op)

Syksy 2005

Opetusohjelman 2005–2006 mukainen sisältö: *Tietämyksen esittäminen sekä sen nojalla tapahtuva päättely ja päätöksenteko. Automaattinen päättely.*

Käytännön asioita

Luennot: maanantaisin klo 12–14, sali TB353
Luennoitsija: ma. prof., TkT Tomi Janhunen, TB335, puh. 451 3255, e-mail: Tomi.Janhunen@hut.fi.

Laskuharjoitukset: tiistaisin klo 15–16, sali TB353
Laskuharjoitusassistentti: DI Emilia Oikarinen, TB345, puh. 451 3292, e-mail: eoikarin@tcs.hut.fi.

Kotisivu: `http://www.tcs.hut.fi/Studies/T-79.5102/`

Uutisryhmä: `opinnot.tik.logiikka`

Sähköposti: `t795102@tcs.hut.fi`

Kurssin sisältö syksyllä 2005

1. **Lauselogiikan toteutusmenetelmät (1. periodi):**
   - Binääriset päätöspuut (BDDt)
   - Davis-Putnam perusalgoritmi ja sen heuristiset muunnelmat
   - Stokastiset menetelmät
2. **Sääntöpohjainen päättely (2. periodi):**
   - Looginen pohja (stabiilit mallit)
   - Approksimoiva päättely (well-founded mallit)
   - Stabiilien mallien laskenta branch and bound -tyyppisellä haulla
   - Ilmaisuvoimaero lauselogiikkaan nähden
3. **Sovelluksia**
   - Digitaalipiirien verifiointi
   - Suunnitteluongelman ratkaiseminen em. menetelmillä

Kurssin suorittaminen

Kurssin suorittaminen edellyttää hyväksyttyjä

- **kolmea kotitehtävää**
  - Ongelman mallintaminen
  - Ratkaisun löytäminen opetetuilla menetelmillä
  - Arvostelu: hyväksytty / hylätty
- **tenttiä**
  - 1. tentti: 21.12.2005 klo 9–12, sali T1
  - lisäksi vuoden 2006 puolella 2 tenttiä (tarpeen mukaan)

Kurssin arvosana määräytyy tentin arvosanan mukaan.

Lauselogiikka 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)

☞ Tietokoneiden suorituskyvyn ja muistin määrän nopea kasvu

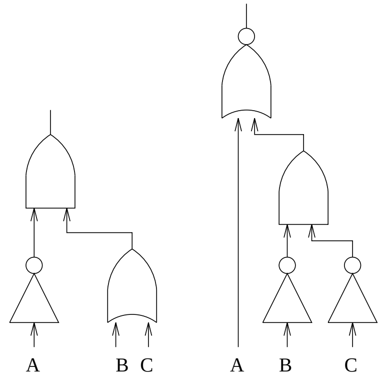
☞ Toteutusmenetelmien kehitys

Sovellutusalueita

- Digitaalipiirien synteesi ja verifiointi
- Vikadiagnoosi
- Testien generointi
- Rinnakkaisten ja hajautettujen järjestelmien verifiointi
- Reitittäminen ja logistiikka
- Suunnittelu
- Erilaiset kombinatoriset ongelmat
- Rajoiteohjelmointi

Esimerkki: digitaalipiirien verifiointi

- Piirin määrittely ja toteutus esitetään logiikan lauseina.
- Toteutuksen oikeellisuus osoitetaan tarkastamalla looginen ekvivalenssi määrittelyn kanssa.



$\neg A \wedge (B \vee C) \equiv \neg(A \vee (\neg B \wedge \neg C))$

Esimerkki: rajoiteohjelmointi

3-väritysongelma: Voidaanko annetun graafin $G = \langle S, K \rangle$ solmut värittää kolmella värillä siten, ettei naapureilla ole sama väri.

Graafin jokaista solmua $a \in S$ kohti 3 atomilausetta: P_a, V_a, S_a

Rajoitteiden (lauselogiikan lauseiden) joukko:

1. Jokaista solmua $a \in S$ kohti lause $P_a \vee V_a \vee S_a$.
2. Jokaista kaarta $\langle a, b \rangle \in K$ kohti lauseet

$$\neg P_a \vee \neg P_b,$$
$$\neg V_a \vee \neg V_b \text{ ja}$$
$$\neg S_a \vee \neg S_b.$$

☞ Graafilla on 3-väritys jos ja vain jos lausejoukko on toteutuva.

Esimerkki: sääntöpohjainen rajoiteohjelmointi

```
kaari(a,b). kaari(b,c). kaari(c,a).
solmu(X) :- kaari(X,Y).
solmu(Y) :- kaari(X,Y).

p(X) :- not v(X), not s(X), solmu(X).
v(X) :- not s(X), not p(X), solmu(X).
s(X) :- not p(X), not v(X), solmu(X).

:- p(X), p(Y), kaari(X,Y).
:- v(X), v(Y), kaari(X,Y).
:- s(X), s(Y), kaari(X,Y).
```

☞ Saadaan kuusi erilaista ratkaisua.