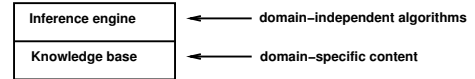


- Tarkastelemme nyt *tietämiskannan* (knowledge base, KB) omaavia agentteja, joita kutsumme *tietämyspohjaisiksi toimijoiksi* (knowledge-based agents)
- Tietämyspohjaiset toimijat käyttävät tietämiskantaa *tiedon esittämiseen* (representation of knowledge) havainnoistaan ja pystyvät tietämyksen avulla *tekemään päätelmiä* (reasoning) maailman tilasta.
- Tämä on erityisen tärkeää osittain havaittavissa ympäristöissä, koska päättely voi antaa tietoa myös ei-havaittavista asioista. Tietojen seurauksien päättely mahdollistaa paremman suorituksen kuin heijastetoimijat.
- Myös ihmiset käyttävät päättelyä ja kielen ymmärtämisessä se on välttämätöntä monimerkityksisyyden takia: puhujan intentio ei ole suoraan havaittavissa.

- Tietämyspohjaiset toimijat ovat joustavia:
 - voivat vaihtaa tehtäviään kun niille annetaan uusia tavoitteita
 - parantavat suoritustaan saadessaan uutta informaatiota ympäristöstä
 - mukautuvat ympäristön muutoksiin päivittämällä tietämystään
- Keskeinen komponentti tietämiskanta on joukko lauseita (sentences), jotka on ilmaistu *tietämyksenesityskielellä* (knowledge representation language)



luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.1/81

luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.2/81

- Tietämiskantaa operoidaan komennoilla TELL ja ASK.
- Kumpikin tehtävä vaatii päättelyä eli uusien lauseiden johtamista vanhoista.
- Perusvaatimus *loogiselle toimijalle* (logical agent): se mitä saadaan vastaukseksi kysymyksestä (ASK) täytyy olla seurausta siitä mitä tietämiskannalle on kerrottu (TELL).

```

function KB-AGENT(percept) returns an action
  static: KB, a knowledge base
         t, a counter, initially 0, indicating time

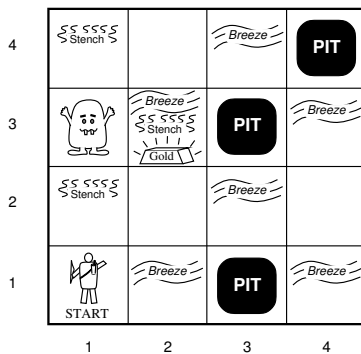
  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
  action ← ASK(KB, MAKE-ACTION-QUERY(t))
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))
  t ← t + 1
  return action

```

luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.3/81

luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.4/81

ESIMERKKIYMPÄRISTÖ: MÖRKÖMAAILMA

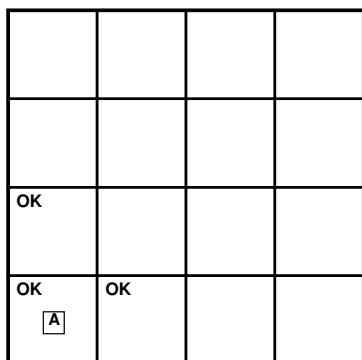


luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.5/81

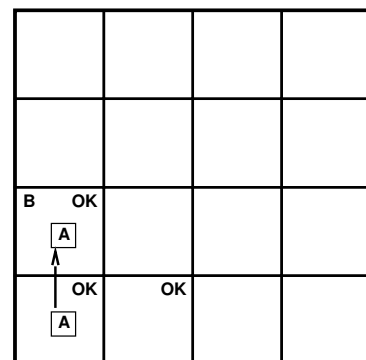
Tämä ympäristö on

- vain osittain havaittava,
- deterministinen,
- sekventiaalinen,
- staattinen,
- diskreetti,
- yhden agentin ympäristö (koska mörkö ei tee mitään).

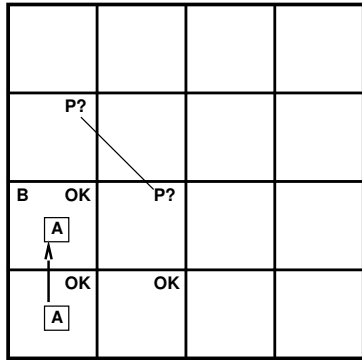
luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.6/81



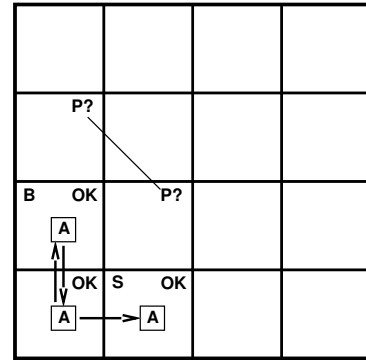
luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.7/81



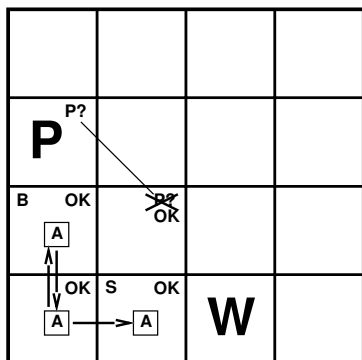
luku07.tex – 58066-7 Tehtävy (4 ov / 8 op) – Rauli Hakli – 13/10/2005 – 1:18 – p.8/81



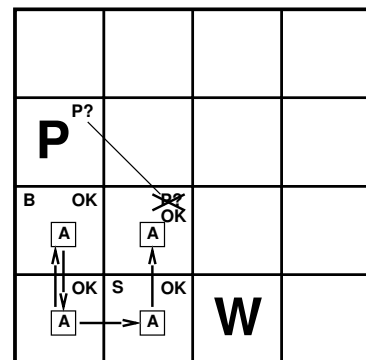
lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.9/81



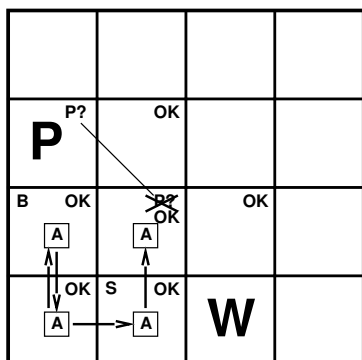
lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.10/81



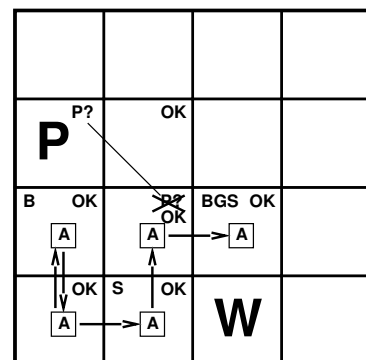
lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.11/81



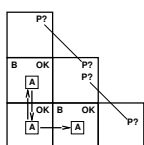
lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.12/81



lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.13/81



lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.14/81



Viimaa ruuduissa (1,2) ja (2,1)
 $\Rightarrow$ ei turvallisia suuntia

Lemua ruudussa (1,1)

$\Rightarrow$ ei turvallista liikkua

Voi käyttää pakotusstrategiaa:

ammu suoraan eteenpäin

mörkö oli siellä $\Rightarrow$ nuoli osui $\Rightarrow$ OK

mörkö ei ollut siellä $\Rightarrow$ OK



LOGIIKKA

- Logiikalla tarkoitetaan formaalia kieltä, jonka avulla voidaan esittää tietämystä ja tehdä päättelyä. Tarvitaan:
 - syntaksi* (kielioppi): aakkosto ja lauseenmuodostussäännöt
 - semantiikka* (merkitysoppi): määrää lauseiden *totuuden* (truth) suhteessa malleihin (mahdollisiin maailmoihin)
- Esimerkiksi aritmetiikan kielessä
 - $x + 2 \geq y$ on lause; $x + 2 + y >$ ei ole lause
 - $x + 2 \geq y$ on tosi joss luku $x + 2$ ei ole pienempi kuin luku y
 - $x + 2 \geq y$ on tosi maailmassa jossa $x = 7, y = 1$
 - $x + 2 \geq y$ on epätosi maailmassa jossa $x = 0, y = 6$
- objektikieli / metakieli

lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.15/81

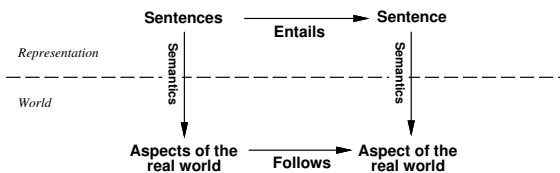
lukut07.tex – 58066-7 Teksti (4 ov / 8 ap) – Raul Hakli – 13/10/2005 – 1:18 – p.16/81

- Loogisella seurauksella tarkoitetaan että jokin lause seuraa toisista lauseista. Ajatellaan KB lausejoukoksi ja sanotaan, että lause p seuraa loogisesti tietämyskannasta KB

$$KB \models p$$

joss kaikissa malleissa, joissa KB on tosi, myös p on tosi.

- Esimerkiksi jos tietämyskannassa on lauseet "HIFK voitti" ja "Jokerit voitti", niin niistä seuraa loogisesti myös lause "HIFK voitti tai Jokerit voitti".
- Esimerkiksi lauseesta $x + y = 4$ seuraa loogisesti lause $4 = x + y$.
- Looginen seuraussuhde vallitsee lauseiden eli syntaktisten elementtien välillä, mutta se perustuu semantiikkaan.

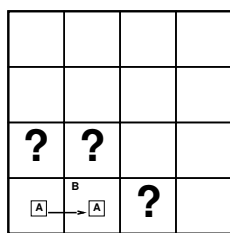


lukut07.tex - 58066-7 Teknolgy (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.18/81

Tilanne kun ruudun [1,1] havainnot on tehty.

Siirrytään oikealle ja havaitaan viimaa ruudussa [2,1]

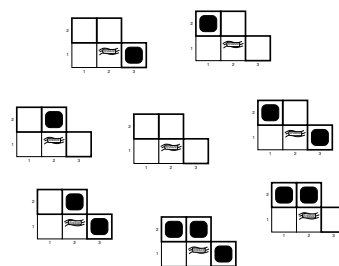
Tarkastellaan kuoppien sijaintia ja konstruoidaan mahdolliset mallit.



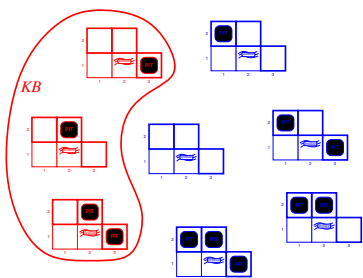
3 totuusarvomuuttujaa $\Rightarrow$ 8 mahdollista mallia

Viereisen ruudun turvallisuus voidaan saada selville mallintarkistuksen (model checking) avulla: Käydään läpi kaikki mahdolliset mallit ja selvitetään, onko lause josta olemme kiinnostuneet, esim. α , tosi kaikissa niissä malleissa, joissa KB on tosi.

lukut07.tex - 58066-7 Teknolgy (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.19/81

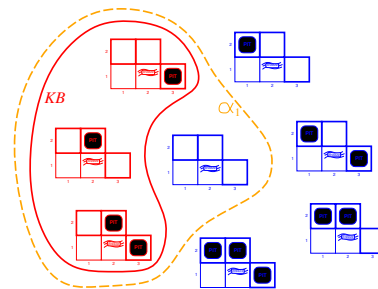


lukut07.tex - 58066-7 Teknolgy (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.20/81



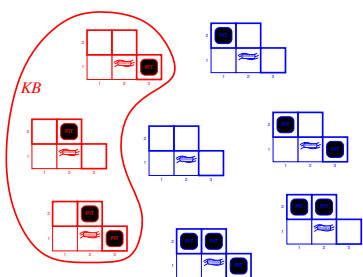
KB = mörkömaailman säännöt + tehdyt havainnot

lukut07.tex - 58066-7 Teknolgy (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.21/81



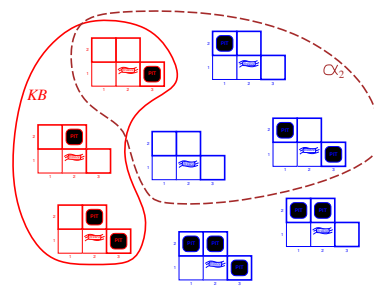
KB = mörkömaailman säännöt + tehdyt havainnot
 $\alpha_1 = "[1,2]$ on turvallinen"
 mallintarkistus osoittaa että $KB \models \alpha_1$

lukut07.tex - 58066-7 Teknolgy (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.22/81



KB = mörkömaailman säännöt + tehdyt havainnot

lukut07.tex - 58066-7 Teknolgy (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.23/81



KB = mörkömaailman säännöt + tehdyt havainnot
 $\alpha_2 = "[2,2]$ on turvallinen"
 $KB \not\models \alpha_2$

lukut07.tex - 58066-7 Teknolgy (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.24/81

Tarkastellaan aluksi *lause*- eli *propositiologiikkaa* (propositional logic)

- Propositiosymbolit $P_1, P_2, \dots$ ovat lauseita (atomilauseet), lisäksi **T** ja **E** ovat lauseita
- Jos S on lause, niin $\neg S$ on lause (negaatio)
- Jos S_1 ja S_2 ovat lauseita, niin $(S_1 \wedge S_2)$ on lause (konjunktio)
- Jos S_1 ja S_2 ovat lauseita, niin $(S_1 \vee S_2)$ on lause (disjunktio)
- Jos S_1 ja S_2 ovat lauseita, niin $(S_1 \rightarrow S_2)$ on lause (implikaatio)
- Jos S_1 ja S_2 ovat lauseita, niin $(S_1 \leftrightarrow S_2)$ on lause (ekvivalenssi)

Sulut voi jättää pois kunhan ei synny monimerkityksisyyttä.
Sidontasäännöt: ensin $\neg$, sitten $\wedge, \vee$ ja lopuksi $\rightarrow, \leftrightarrow$

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.25/81

Kukin malli kiinnittää atomilauseiden totuusarvon, lisäksi **T** on tosi ja **E** epätosi
Yhdistetyt lauseet evaluoidaan suhteessa malliin m seuraavasti:

- $\neg S$ on tosi, joss S ei ole tosi.
- $S_1 \wedge S_2$ on tosi, joss S_1 on tosi ja S_2 on tosi.
- $S_1 \vee S_2$ on tosi, joss S_1 on tosi tai S_2 on tosi.
- $S_1 \rightarrow S_2$ on tosi, joss S_1 on epätosi tai S_2 on tosi.
- $S_1 \leftrightarrow S_2$ on tosi, joss $S_1 \rightarrow S_2$ on tosi ja $S_2 \rightarrow S_1$ on tosi.

Totuustaulut:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
E	E	T	E	E	T	T
E	T	T	E	T	T	E
T	E	E	E	T	E	E
T	T	E	T	T	T	T

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.26/81

- Propositiologiikan semantiikka antaa säännöt lauseiden totuusarvon määrittämiseksi mallien suhteen.
- Olkon esimerkiksi malli $M = \{P_{1,2} = \mathbf{E}, P_{2,2} = \mathbf{E}, P_{3,1} = \mathbf{T}\}$
- Nyt lauseen $\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1})$ totuusarvo mallissa M voidaan laskea rekursiivisesti konnektiivien totuustaulujen avulla:
 $\mathbf{T} \wedge (\mathbf{E} \vee \mathbf{T}) = \mathbf{T}$
- Tietämiskanta KB , joka on muodostettu operaatioin $\text{TELL}(KB, S_1), \dots, \text{TELL}(KB, S_n)$ on lauseiden $S_1, \dots, S_n$ konjunktio: $KB = S_1 \wedge \dots \wedge S_n$.
- KB :tä voidaan siis käsitellä loogisena lauseena.

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.27/81

Nyt voimme koodata esim. Mörkömaailman tietämyksen:

- Valitaan sopivat propositiosymbolit ja niiden tulkinnat:
 $P_{i,j}$ tarkoittaa että ruudussa $[i, j]$ on kuoppa
 $B_{i,j}$ tarkoittaa että ruudussa $[i, j]$ on viimaa
:
Koodataan yleiset linalaisuuudet, kuten se että ruudussa on viimaa jos ja vain jos viereisessä ruudussa on kuoppa:
 $\neg P_{1,1} \quad (R_1)$
 $B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1}) \quad (R_2)$
 $B_{2,1} \leftrightarrow (P_{1,3} \vee P_{2,2} \vee P_{3,1}) \quad (R_3)$
:
Kun tehdään havaintoja ympäristöstä, lisätään niitä vastaavat positiiviset tai negatiiviset literaalit:
 $\neg B_{1,1} \quad (R_4)$
 $B_{2,1} \quad (R_5)$

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.28/81

- Kun kaikki tämä on saatu syötettyä tietämiskantaan TELL -komennon avulla, voidaan ryhtyä kyselemään ASK -komennon avulla pätekö $KB \models \alpha$ jollekin α .
- Olisi esimerkiksi hyödyllistä tietää, seuraako tietämyksestäme $\neg P_{1,2}$.
- Esimerkissämme kahden ensimmäisen ruudun vierailun aikana relevantteja propositiosymboleja on 7 kpl, joten mahdollisia malleja on $2^7 = 128$ kpl.
- KB on tosi vain kolmessa näistä.
- Mallintarkistusalgoritilla näemme että $\neg P_{1,2}$ on tosi kaikissa näissä kolmessa mallissa, joten ruudussa $[1, 2]$ ei ole kuoppaa.
- Sen sijaan $P_{2,2}$ on tosi kahdessa ja epätosi yhdessä KB :n mallissa, joten emme voi vielä päätellä ruudun $[2, 2]$ kuoppaisuutta.

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.31/81

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
E	E	E	E	E	E	E	T	T	T	T	E	E
E	E	E	E	E	E	T	T	T	E	T	E	E
:	:	:	:	:	:	:	:	:	:	:	:	:
E	T	E	E	E	E	E	T	T	E	T	T	E
E	T	E	E	E	E	T	T	T	T	T	T	T
E	T	E	E	E	T	T	T	T	T	T	T	T
E	T	E	E	T	E	E	T	E	E	T	T	E
:	:	:	:	:	:	:	:	:	:	:	:	:
T	T	T	T	T	T	T	E	T	T	E	T	E

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.30/81

```

function TT-ENTAILS?(KB,  $\alpha$ ) returns true or false
  inputs: KB, the knowledge base, a sentence in propositional logic
          $\alpha$ , the query, a sentence in propositional logic

  symbols  $\leftarrow$  a list of the proposition symbols in KB and  $\alpha$ 
  return TT-CHECK-ALL(KB,  $\alpha$ , symbols, [])

function TT-CHECK-ALL(KB,  $\alpha$ , symbols, model) returns true or false
  if EMPTY?(symbols) then
    if PL-TRUE?(KB, model) then return PL-TRUE?( $\alpha$ , model)
    else return true
  else do
     $P \leftarrow \text{FIRST}(\text{symbols})$ ;  $\text{rest} \leftarrow \text{REST}(\text{symbols})$ 
    return TT-CHECK-ALL(KB,  $\alpha$ , rest, EXTEND( $P$ , true, model)) and
           TT-CHECK-ALL(KB,  $\alpha$ , rest, EXTEND( $P$ , false, model))

```

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.31/81

- Tämä mallintarkistukseen perustuva syvyysuuntainen algoritmi on eheä, koska se perustuu suoraan loogisen seuraamisen määrittelmään.
- Se on myös täydellinen, koska se toimii mille tahansa KB :lle ja α :lle aina pysähtyen. Mahdollisia malleja on äärellinen määrä.
- Jos KB ja α sisältävät yhteensä n symbolia, niin malleja on 2^n kpl, siis aikavaatimus on luokkaa $O(2^n)$.
- Itse asiassa kaikkien tunnettujen lauselogiikan päättelyalgoritmien pahimman tapauksen aikavaativuus on eksponentiaalinen syötteen koon suhteen.
- Lauselogiikan loogisen seuraamisen päätösongelma on co-NP-täydellinen.

lukut7.tex – 58066-7 Tekijä (4 ov / 8 opp) – Raul Hakli – 13/10/2005 – 1:18 – p.32/81

- Lause β seuraa loogisesti lauseesta α , merkitään $\alpha \models \beta$, joss kaikissa malleissa, joissa α on tosi, myös β on tosi.
- Lauseet α ja β ovat loogisesti ekvivalentteja, merkitään $\alpha \equiv \beta$, joss $\alpha \models \beta$ ja $\beta \models \alpha$.
- Lause on loogisesti tosi / validi / pätevä / tautologia, jos se on tosi kaikissa malleissa, siis $\models \alpha$.
- Lause on toteutuva (satisfiable), jos on olemassa malli, jossa se on tosi (malli toteuttaa sen).
- Lause on kumoutuva (refutable), jos on olemassa malli, jossa se on epätosi.
- Lause on loogisesti epätosi / ristiriitainen (unsatisfiable), jos se on epätosi kaikissa malleissa.
- Yhteys loogisen seurauksen ja materiaallisen implikaation välillä: deduktioteoreema: $\alpha \models \beta$ joss $(\alpha \rightarrow \beta)$ on loogisesti tosi.

lukut07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.34/81

- Lause α on johdettavissa (derivable) tietämuskannasta tai teoriasta KB jollain todistusjärjestelmällä i

$$KB \vdash_i \alpha$$

jos se voidaan todistaa KB :n lauseista käyttäen i :n päättelysääntöjä ja aksioomia.

- Lause α on todistuva eli teoreema, jos $\vdash_i \alpha$ eli se on johdettavissa tyhjästä lausejoukosta.
- Päättelyjärjestelmä i on eheä (sound), joss siitä että $KB \vdash_i \alpha$ seuraa että $KB \models \alpha$, siis jos sillä voidaan todistaa ainoastaan tietämuskannan loogisia seurauksia.
- Päättelyjärjestelmä i on täydellinen (complete), joss siitä että $KB \models \alpha$ seuraa että $KB \vdash_i \alpha$, siis jos sillä voidaan todistaa kaikki tietämuskannan loogiset seuraukset.

lukut07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.34/81

PÄÄTTELY PROPOSITIOLOGIIKASSA

- Päättelyjärjestelmän avulla voimme tutkia, mitkä lauseet ovat todistuvia tietämuskannan perusteella.
- Tarvitaan päättelysääntöjä (inference rules), kuten esim.

$$\text{Modus ponens: } \frac{\alpha \rightarrow \beta \quad \alpha}{\beta}$$

$$\text{Konjunktion poisto: } \frac{\alpha \wedge \beta}{\alpha} \quad \frac{\alpha \wedge \beta}{\beta}$$

$$\text{Konjunktion tuonti: } \frac{\alpha \quad \beta}{\alpha \wedge \beta}$$

- Nämä säännöt ovat aina eheitä joten niitä voidaan soveltaa suoraan eikä ole tarvetta tehdä mallintarkistusta.
- Eheitä päättelysääntöjä saadaan esimerkiksi seuraavista tunnetuista ekvivalensseista.

lukut07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.35/81

Tunnettuja loogisia ekvivalensseja:

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	kommutatiivisuus
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	kommutatiivisuus
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	assosiatiivisuus
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	assosiatiivisuus
$\neg(\neg\alpha) \equiv \alpha$	kaksoisnegaation poisto
$(\alpha \rightarrow \beta) \equiv (\neg\beta \rightarrow \neg\alpha)$	kontrapositio
$(\alpha \rightarrow \beta) \equiv (\neg\alpha \vee \beta)$	implikaation poisto
$(\alpha \leftrightarrow \beta) \equiv ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$	ekvivalenssin poisto
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	de Morgan
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	de Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributiivisuus
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributiivisuus

lukut07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.37/81

Tavallisia päättelyjärjestelmiä:

- Hilbert-järjestelmät
- Gentzenin luonnollisen päättelyn järjestelmät
- Gentzenin sekvenssikalkyyli
- Resoluuioon perustuvat järjestelmät

lukut07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.38/81

ESIMERKKI PÄÄTTELYSTÄ

- Tarkastellaan päättelyä mörkömaailmassa: pyritään todistamaan ettei ruudussa $[1, 2]$ ole kuoppaa.
- Olkoon tilanne seuraavanlainen:

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1	2,1	3,1	4,1
OK	OK		

(a)

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	p?	4,2
OK			
1,1	2,1	3,1	p?
v	OK	OK	4,1

(b)

$\square$ = Agent
 $\circ$ = Breeds
 $\circ$ = Glitter, Gold
 $\square$ = Safe square
 $\circ$ = Pit
 $\circ$ = Stench
 $\circ$ = Visited
 $\circ$ = Wumpus

Tietämuskannassa on lauseet $R_1, \dots, R_5$:

$\neg P_{1,1}$	(premissi) (R_1)
$B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})$	(premissi) (R_2)
$B_{2,1} \leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$	(premissi) (R_3)
$\neg B_{1,1}$	(premissi) (R_4)
$B_{2,1}$	(premissi) (R_5)
$(B_{1,1} \rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$	$(\leftrightarrow ER_2) (R_6)$
$(P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1}$	$(\wedge ER_6) (R_7)$
$(\neg B_{1,1} \rightarrow \neg(P_{1,2} \vee P_{2,1}))$	(kontrap. $R_7) (R_8)$
$\neg(P_{1,2} \vee P_{2,1})$	(MP $R_8, R_4) (R_9)$
$\neg P_{1,2} \wedge \neg P_{2,1}$	(de Morgan $R_9) (R_{10})$

Löydettiin siis todistus (proof) sille, ettei ruuduissa $[1, 2]$ ja $[2, 1]$ ole kuoppia.

lukut07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.39/81

lukut07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.40/81

- Koska päättely propositiologiikassa on NP-täydellistä, niin pahimmassa tapauksessa todistuksen etsiminen on yhtä tehontonta kuin mallintarkastus.
- Käytännössä kuitenkin irrelevanttien tietojen huomiotta jättäminen tekee usein todistamisen tehokkaaksi.
- Esimerkiksi edellisessä todistuksessa ei tarvinnut huomioida lainkaan säännöissä R_1 , R_3 ja R_5 mainittuja proposiitosymboleita $B_{2,1}$, $P_{1,1}$, $P_{2,2}$ ja $P_{3,1}$ kun taas totuustaulukkomenetelmässä nekin olisi pitänyt ottaa mukaan tarkasteluun.
- Logiikan *monotonisuus*: tiedon lisääntyessä loogisten seurausten määrä voi vain kasvaa: jos $KB \models \alpha$, niin $KB \wedge \beta \models \alpha$.
- Edellä mainituissa päättelyjärjestelmissä yhdenkin säännön poistaminen rikkoo järjestelmän täydellisyyden.

lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.41/81

- Resoluutiosääntö yhdistettynä täydelliseen hakualgoritmiin antaa (refutaatio)täydellisen päättelyjärjestelmän.

- Resoluutioaskel kahden literaalin tapauksessa:

$$\frac{l_1 \vee l_2 \quad \neg l_2 \vee l_3}{l_1 \vee l_3}$$

- Tarkastellaan jälleen päättelyä Mörkömaailmassa:

1,4	2,4	3,4	4,4
1,3 w!	2,3	3,3	4,3
1,2 $\begin{bmatrix} A \\ S \\ OK \end{bmatrix}$	2,2	3,2	4,2
1,1 $\begin{bmatrix} V \\ OK \end{bmatrix}$	2,1 $\begin{bmatrix} B \\ V \\ OK \end{bmatrix}$	3,1 p!	4,1

(a)

1,4	2,4 p?	3,4	4,4
1,3 w!	2,3 $\begin{bmatrix} A \\ S \\ G \\ B \end{bmatrix}$	3,3 p?	4,3
1,2 $\begin{bmatrix} S \\ V \\ OK \end{bmatrix}$	2,2 $\begin{bmatrix} V \\ B \\ OK \end{bmatrix}$	3,2	4,2
1,1 $\begin{bmatrix} V \\ OK \end{bmatrix}$	2,1 $\begin{bmatrix} B \\ V \\ OK \end{bmatrix}$	3,1 p!	4,1

(b)

lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.42/81

- Ruudussa $[1, 2]$ agentti aistii löyhkän muttei viimaa:

$$\neg B_{1,2} \quad (R_{11})$$

$$B_{1,2} \leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3}) \quad (R_{12})$$
- Vastaavasti kuin aiemmin, voidaan todeta, ettei ruuduissa $[2, 2]$ ja $[1, 3]$ ole kuoppia:

$$\neg P_{2,2} \quad (R_{13})$$

$$\neg P_{1,3} \quad (R_{14})$$
- Ekvivalenssin poisto (R_3) ja MP (R_5) tuottavat disjunktion

$$P_{1,1} \vee P_{2,2} \vee P_{3,1} \quad (R_{15})$$
- Nyt voidaan soveltaa resoluutiosääntöä literaalilla $\neg P_{2,2}$ (R_{13}):

$$P_{1,1} \vee P_{3,1} \quad (R_{16})$$
- Ja sitten literaalilla $\neg P_{1,1}$ (R_1):

$$P_{3,1} \quad (R_{17})$$

lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.43/81

- Äsken käytettiin *yksikköresoluutiota* (unit resolution):

$$\frac{l_1 \vee \dots \vee l_k \quad m}{l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_k}$$

missä l_i ja m ovat toistensa komplementtiliteraaleja, siis toinen on atomilause ja toinen sen negaatio.

- Yleinen resoluutiosääntö on seuraava:

$$\frac{l_1 \vee \dots \vee l_k \quad m_1 \vee \dots \vee m_n}{l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n}$$

missä l_i ja m_j ovat toistensa komplementtiliteraaleja.

- Sääntöä sovellettaessa johtopäätös voi sisältää saman literaalin kaksi kertaa, jolloin toinen esiintymä täytyy poistaa kuten alla:

$$\frac{A \vee B \quad A \vee \neg B}{A}$$

lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.44/81

- Resoluutio on eheä päättelysääntö ja se on myös täydellinen siinä mielessä, että sen avulla voidaan ratkaista mielivaltaisesta propositiologiikan lauseesta, seuraako lause tietokannasta vai ei. Tätä kutsutaan *refutaatiotäydellisyydeksi*.
- Resoluutiolla ei siis voi tuottaa automaattisesti kaikkia tietokannasta seuraavia lauseita. Esimerkiksi A :sta ei resoluutiolla saada pääteltyä $A \vee B$. Sen sijaan voidaan kyllä testata seuraako $A \vee B$ lauseesta A .
- Resoluutiomenetelmän idea perustuu todistustapaan nimeltä *reductio ad absurdum*: kun halutaan todistaa, että jokin lause α seuraa loogisesti tietokannasta KB , voidaan muodostaa lause $KB \wedge \neg \alpha$ ja johtaa siitä ristiriita.
- Kutsutaan myös nimillä refutation tai proof by contradiction eli epäsuora todistus tai todistaminen ristiriidan kautta.
- Perustelu: $\alpha \models \beta$ joss $\alpha \wedge \neg \beta$ on loogisesti epätos.

lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.47/81

Resoluutiomenetelmän käyttö vaatii ensin lauseiden muuntamista *konjunkttiiviseen normaalimuotoon* (CNF) (joka on klausuulien konjunktio): $(l_{1,1} \vee \dots \vee l_{1,k}) \wedge \dots \wedge (l_{n,1} \vee \dots \vee l_{n,m})$. Esim. $B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})$

- Eliminoidaan ekvivalenssit:

$$(B_{1,1} \rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$$
- Eliminoidaan implikaatit:

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg (P_{1,2} \vee P_{2,1}) \vee B_{1,1})$$
- Siirretään negaatit mahdollisimman syvälle sisään kaksoisnegaation poistolla ja de Morganin säännöillä:

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$$
- Sovelletaan distributiivisuutta tarvittaessa:

$$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.48/81

Algoritmi:

- Muutetaan lause $(KB \wedge \neg \alpha)$ CNF-muotoon.
- Käytetään saatuja lauseita premisseinä resoluutiosäännölle ja lisätään johtopäätökset lauseiden joukkoon.
- Jatketaan kunnes joko
 - Uusia lauseita ei enää voi lisätä $\Rightarrow \alpha$ ei seuraa tietokannasta.
 - Resoluutiosäännön soveltaminen tuottaa ristiriidan eli tyhjän disjunktion $\Rightarrow \alpha$ seuraa tietokannasta.

```

function PL-RESOLUTION( $KB, \alpha$ ) returns true Or false
inputs:  $KB$ , the knowledge base, a sentence in propositional logic
 $\alpha$ , the query, a sentence in propositional logic

clauses  $\leftarrow$  the set of clauses in the CNF representation of  $KB \wedge \neg \alpha$ 
new  $\leftarrow \{ \}$ 
loop do
  for each  $C_i, C_j$  in clauses do
    resolvents  $\leftarrow$  PL-RESOLVE( $C_i, C_j$ )
    if resolvents contains the empty clause then return true
  new  $\leftarrow$  new  $\cup$  resolvents
if new  $\subseteq$  clauses then return false
clauses  $\leftarrow$  clauses  $\cup$  new

```

lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.47/81

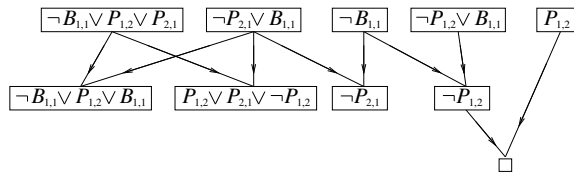
lukut07.tex – 58066-7 Tekniikan (4 ov / 8 op) – Rauli Hakki – 13/10/2005 – 1:18 – p.48/81

$$KB = (B_{1,1} \leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$$

$$\alpha = \neg P_{1,2}$$

Tarkastellaan siis seuraavia konjunktivisessa normaalimuodossa olevia lauseita:

$(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}), (\neg P_{1,2} \vee B_{1,1}), (\neg P_{2,1} \vee B_{1,1}), \neg B_{1,1}$ ja $P_{1,2}$.



Saadaan ristiriita, joten $KB \models \alpha$.

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.49/81

- Resoluutio on täydellinen, mutta siksi myös NP-täydellinen. Käytännön tilanteissa on parempi tyytyä heikompaan mutta tehokkaampaan menetelmään.
- Rajoitetaan päättelyä siten, ettei käsitellä kaikkia lauseita vaan ainoastaan nk. *Hornin lauseita* (Horn clauses): Ne ovat disjunktioita literaaleista, joista korkeintaan yksi on positiivinen ja muut negatiivisia.
- Tällaiset muotoa $\neg P_1 \vee \neg P_2 \vee \dots \vee \neg P_n \vee Q$ olevat lauseet voidaan esittää loogisesti ekvivalentissa muodossa $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$, jossa implikaation etujäsentä kutsutaan lauseen *rungoksi* (body) ja takajäsentä *kärjeksi* (head).
- Jos Hornin lauseella ei ole lainkaan runkoa, niin kyseessä on *fakta* (fact).

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.50/81

- Tällaiset säännöt ovat helposti ymmärrettäviä ja niiden avulla voidaan kuvata monen erikoisalueen tietämystä.
- Päättely Hornin lauseilla voi perustua joko *eteenpäin ketjutukseen* (forward chaining) tai *taaksepäin ketjutukseen* (backward chaining).
- Loogisen seuraamisen tutkimisen aikavaativuus on Hornin lauseilla vain lineaarista tietämyskannan koon suhteen.
- Eteenpäin ketjutuksessa tutkitaan, minkä kaikkien sääntöjen rungot ovat tosia tunnettujen faktojen perusteella. Niiden kärjet voidaan sitten lisätä uusiksi faktoiksi tietämyskantaan. Tätä jatketaan kunnes haluttu johtopäätös on saatu todistettua tai kun uusia päätelmiä ei enää voida tehdä.

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.51/81

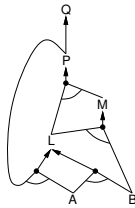
```
function PL-FC-ENTAILS?(KB, q) returns true or false
inputs: KB, a set of propositional Horn clauses
       q, the query, a proposition symbol
local variables: count, a table, indexed by clause, in. # of premises
                inferred, a table, indexed by symbol, each false
                agenda, a list of symbols, initially the symbols in KB

while agenda is not empty do
  p ← POP(agenda)
  unless inferred[p] do
    inferred[p] ← true
    for each Horn clause c in whose premise p appears do
      decrement count[c]
      if count[c] = 0 then do
        if HEAD[c] = q then return true
        PUSH(HEAD[c], agenda)
return false
```

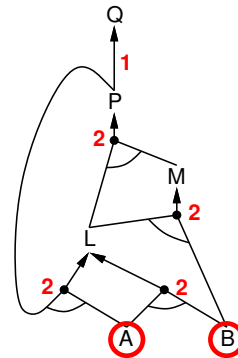
luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.52/81

Algoritmin toimintaa voi havainnollistaa *JA-TAI-verkon* (AND-OR graph) avulla:

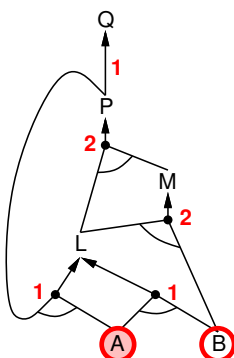
$P \rightarrow Q$
 $L \wedge M \rightarrow P$
 $B \wedge L \rightarrow M$
 $A \wedge P \rightarrow L$
 $A \wedge B \rightarrow L$
 A
 B



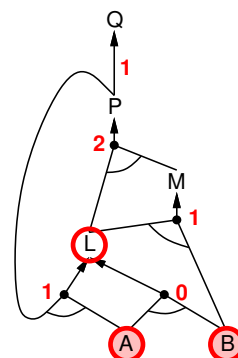
luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.53/81



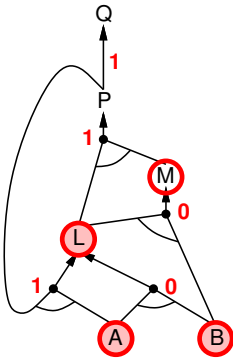
luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.54/81



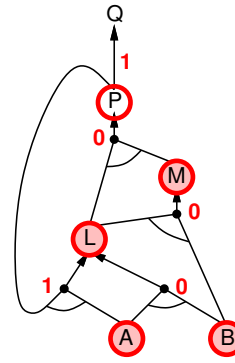
luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.55/81



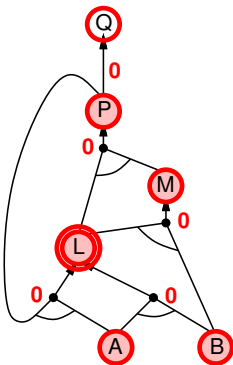
luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.56/81



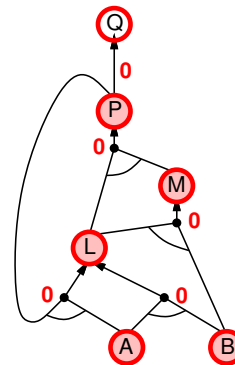
luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.57/81



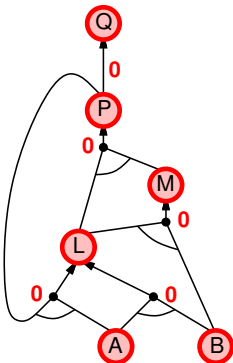
luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.58/81



luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.59/81



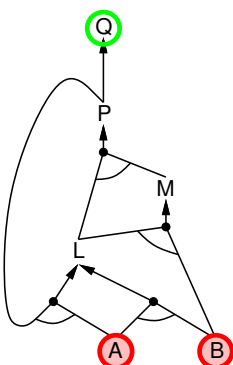
luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.60/81



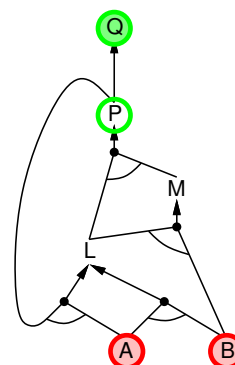
luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.61/81

- Päättely on eheää ja myös täydellistä siinä mielessä, että sen avulla voidaan johtaa kaikki *atomilauseet* jotka seuraavat tietämuskannasta.
- Kyseessä on *datalähtöinen* (data driven) päättely: siinä lähdetään liikkeelle tietämuskannan datasta ja todistetaan kaikkea mahdollista kunnes osutaan kiinnostavaan johtopäätökseen.
- *Tavoiteohjattua* (goal-directed) päättelyä edustaa *taaksepäinjetjutus* (backward chaining), jossa lähdetään liikkeelle halutusta johtopäätöksestä ja tutkitaan mitä täytyy todistaa johtopäätöksen todistamiseksi. Tätä jatketaan kunnes joko päästään faktoihin tai huomataan ettei johtopäätös todistu.
- Tämäkin toimii lineaarisessa ajassa ja eroaa eteenpäinjetjutuksesta siinä, että se käsittelee vain relevanteja faktoja eikä tuota sokeasti kaikkia.

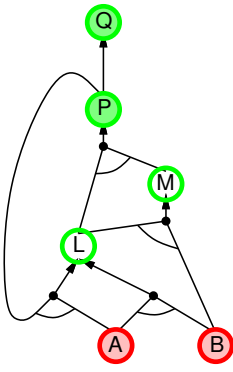
luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.62/81



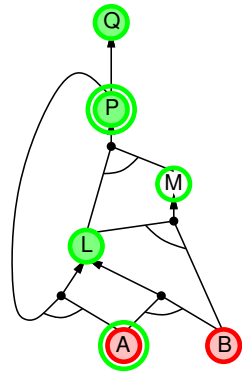
luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.63/81



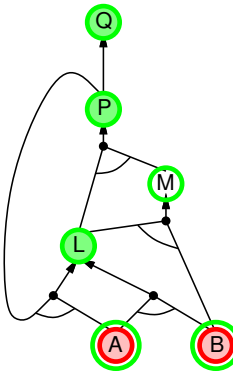
luku07.tex - 58066-7 Tekijäily (4 ov / 8 op) - Raul Hakli - 13/10/2005 - 1:18 - p.64/81



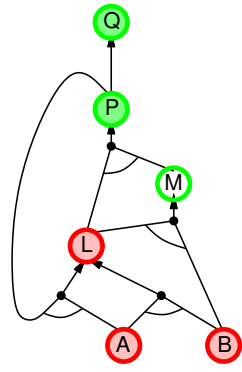
lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.65/81



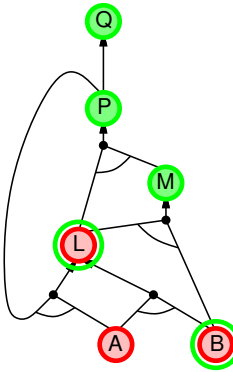
lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.66/81



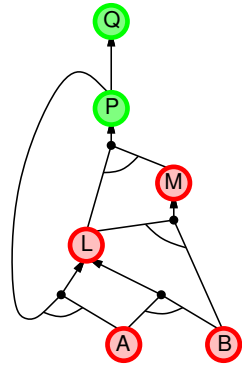
lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.67/81



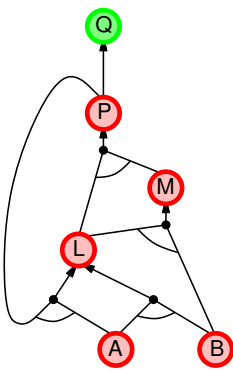
lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.68/81



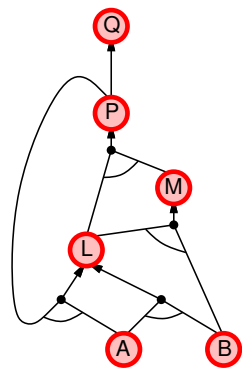
lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.69/81



lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.70/81



lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.71/81



lukut07.tex – 58066-7 Teknolgy (4 cv / 8 op) – Raul Hakli – 13/10/2005 – 1:18 – p.72/81

- Tarkastellaan vielä kahta mallintarkistukseen perustuvaa (semanttista) päättelymekanismia: Davis-Putnam-algoritmi (DPLL) ja WALKSAT-algoritmi, jotka yrittävät löytää mallin jossa annettu lause on tosi, siis toteutusongelmaa (SAT), joka tiedetään NP-täydelliseksi.
- Davis-Putnam-algoritmi
 - Ottaa syötteenään konjunktioivissa normaalimuodossa olevan lauseen.
 - Perustuu peruuttavaan syvyysuuntaiseen etsintään, jossa käydään läpi mahdolliset mallit ja tutkitaan toteuttaako jokin niistä annetun lauseen.
 - Käyttää erilaisia kikkoja haarautumisasasteen pienentämiseen ja haun lopettamiseen mahdollisimman nopeasti (early termination, pure symbol heuristic, unit clause heuristic).
 - Nopeimpia toteutusongelman ratkaisevia algoritmeja.

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.73/81

```

function DPLL-SATISFIABLE?(s) returns true or false
inputs: s, a sentence in propositional logic
clauses ← the set of clauses in the CNF representation of s
symbols ← a list of the proposition symbols in s
return DPLL(clauses, symbols, [])

```

```

function DPLL(clauses, symbols, model) returns true or false
if every clause in clauses is true in model then return true
if some clause in clauses is false in model then return false
P, value ← FIND-PURE-SYMBOL(symbols, clauses, model)
if P is non-null then return DPLL(clauses, symbols-P, [P = value|model])
P, value ← FIND-UNIT-CLAUSE(clauses, model)
if P is non-null then return DPLL(clauses, symbols-P, [P = value|model])
P ← FIRST(symbols); rest ← REST(symbols)
return DPLL(clauses, rest, [P = true|model]) or
       DPLL(clauses, rest, [P = false|model])

```

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.74/81

PAIKALLISEEN ETSINTÄÄN PERUSTUVAT ALGORITMIT

- aloitetaan satunnaisesta totuusarvojakelusta ja muutetaan sitä minimoimalla epätosien klausuulien (clause) lukumäärää (vrt. min-conflicts heuristic)
- paikallisia minimejä on paljon, joten tarvitaan satunnaisuuttakin
- WALKSAT-algoritmi valitsee jonkin toteutumattoman klausuulin ja vaihtaa jonkin siinä esiintyvän literaalien totuusarvon: literaali valitaan joko satunnaisesti tai minimoimalla epätosia klausuuleja.
- Jos WALKSAT löytää mallin, niin syötelause on toteutuva, mutta jos se ei palauta mallia, niin se tarkoittaa vain ettei se kyennyt löytämään sellaista, eikä sitä ettei sellaista ole! Iterointia voidaan tuki jatkaa äärettömän kauan löytämisen varmistamiseksi, mutta silloin algoritmi ei loogisesti epätosien lauseiden kohdalla pysähdy koskaan. → toteutumattomuutta (eli ristiriitaisuutta) ei voi tarkistaa

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.75/81

```

function WALKSAT(clauses, p, max-flips) returns a satisfying model or failure
inputs: clauses, a set of clauses in propositional logic
p, the probability of choosing to do a "random walk" move, typically around 0.5
max-flips, number of flips allowed before giving up

model ← a random assignment of true/false to the symbols in clauses
for i = 1 to max-flips do
  if model satisfies clauses then return model
  clause ← a randomly selected clause from clauses that is false in model
  with probability p flip the value in model of a randomly selected symbol from clause
  else flip whichever symbol in clause maximizes the number of satisfied clauses
return failure

```

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.76/81

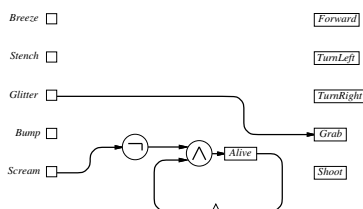
LAUSELOGIIKKAAN PERUSTUVAT AGENTIT

- Koska lauseiden totuutta koskevat ongelmat ovat NP-täydellisiä, käytännön ongelmissa lauseen p totuuden tai epätotuuden selvittämiseksi toimitaankin usein rinnakkaisesti: samaan aikaan käytetään mallintarkistajaa etsimään mallia p :lle ja teoreemantodistajaa etsimään todistusta $\neg p$:lle (jolloin nähtäisiin ettei mallia voi olla olemassa). (Tai vastaavasti jos halutaan tutkia onko p todistuva, käytetään teoreemantodistajaa p :lle ja mallintarkistajaa $\neg p$:lle.)
- Ongelmana lauseiden suuri lukumäärä tietämyksen esittämisessä
- Ajan ja paikan (muutoksen) huomioimattaminen pahentaa tilannetta entisestään.
- Yksi mahdollisuus on koodata logiikka laitteistolle. Tätä käsitellään lyhyesti seuraavaksi.
- Toinen mahdollisuus on siirtyminen ilmaisuvoimaisemman kielen käyttämiseen. Tätä käsitellään seuraavassa luvussa.

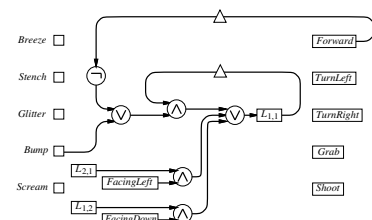
luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.77/81

LOGIikkAPIIREIHIN PERUSTUVAT AGENTIT

- loogiset konnektiivit esitetään porteilla (gates) ja rekisterit koodaavat propositionien totuusarvot. Tulosteina piireillä on rekisteriarvot, jotka tarkoittavat tekoja.
- ajan käsittely on helpompaa kuin tavallisessa deklarativisessa esityksessä, koska esitystä ei tarvitse kopioida jokaiselle ajanhetkelle vaan piiri vain evaluodaan kyseisellä ajanhetkellä



luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.78/81



- Ongelma: kaikilla rekistereillä täytyy aina olla arvot (tosi tai epätosi) kun taas tietämyskannassa kullakin atomilauseella voi olla kolme tilaa: atomilause on kannassa, atomilauseen negatio on kannassa tai kumpikin puuttuu. Ongelma voidaan kiertää käyttämällä "tietopropositioita" (knowledge propositions).
- Muita vaatimuksia piirien käytölle: lokaalisuus, syklittömyys

luk07.tex – 58066-7 Teknolgy (4 ov / 8 op) – Raul Hakki – 13/10/2005 – 1:18 – p.80/81

Esityksen tiiviys: Piiripohjainen esitys ei vaadi samojen asioiden esittämistä jokaiselle ajanhetkelle erikseen, mutta siinäkin jokaiselle paikalle täytyy olla oma piirinsä

Laskennallinen tehokkuus: Päättely voi pahimmassa tapauksessa viedä eksponentiaalisen ajan symboleiden lukumäärään nähden, kun taas piirit voidaan evaluoida lineaarisessa ajassa niiden kokoon nähden. Tosin DPLL toimii yleensä nopeammin ja kaikki päättelyt jotka piiripohjaiset agentit tekevät, voidaan hoitaa myös DPLL:llä lineaarisessa ajassa.

Täydellisyys: Periaatteessa piireillä voidaan tehdä kaikki mitä tietämyskantoihin perustuvilla ohjelmillakin, mutta piirit kasvavat eksponentiaalisen suuriksi tietämyskantoihin verrattuna ja hyvin monimutkaisiksi. Tietämyskannalla voidaan tehdä sellaisiakin päättelyitä joita ei ole etukäteen erikseen ohjelmoitu.

Konstruoinnin helppous: Deklaratiivista pidetään usein helpompana.