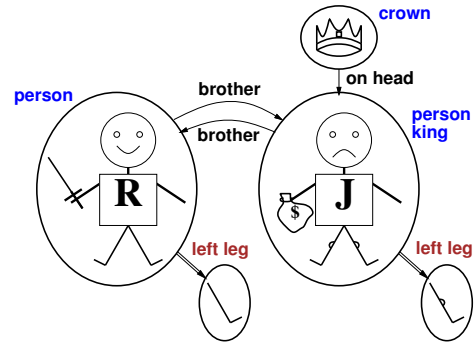


- Lauselogiikassa propositiot ilmaistaan atomilauseiden ja konnektiivien avulla: nyt tarkastellaan atomilauseita yksityiskohtaisemmin ja sallitaan niille rakennetta.
- Oletetaan että maailma koostuu yksilöolioista, joilla on ominaisuuksia ja suhteita (relations).
- Tarkemmin sanoen relaatiot voivat olla yksipaikkaisia, eli ominaisuuksia (joskus yksipaikkaisia relaatioita sanotaan predikaateiksi, joskus monipaikkaisiakin kutstutaan predikaateiksi). Relaatiot voivat myös toteuttaa funktionaalisuusehdon, jolloin niitä voidaan tarkastella funktioina.

luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.1/12



luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.2/12

SYNTAKSI JA SEMANTIikka

- Symbolit ja tulkinta: *vakiosymbolit* viittaavat olioihin, *predikaattisymbolit* relaatioihin ja *funktiosymbolit* funktioihin. *Tulkinta* (interpretation) sitoo symbolit maailman vastineisiinsa.
- Maailma ja symbolien tulkinta muodostavat predikaattilogiikan mallin.
 - Esimerkkikuvaan liittyvä tulkinta voisi nyt olla vaikka sellainen, jossa sana *Richard* viittaa Rikhard Leijonamieleen, *John* viittaa Juhana Maattomaan, *Brother* viittaa joukkoon $\{ \langle \text{Rikhard Leijonamieli}, \text{Juhana Maaton} \rangle, \langle \text{Juhana Maaton}, \text{Rikhard Leijonamieli} \rangle \}$, jne.
 - Toisaalta tulkinta voisi olla sellainen, jossa *Richard* viittaa kruunuun ja *John* Juhanan vasempaan jalkaan tai vaikkapa siihen samaan kruunuun kuin *Richard*.

luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.3/12

- Termi $f(t_1, \dots, t_n)$ viittaa siihen oloon, joka saadaan ottamalla termien $t_1, \dots, t_n$ viittaamat oliot $d_1, \dots, d_n$ ja soveltamalla niihin sitä funktiota F , joka on $f:n$ tulkinta.
- Atomilause $P(t_1, \dots, t_n)$ on tosi mikäli oliot, joihin termit $t_1, \dots, t_n$ viittaavat ovat keskenään siinä relaatiossa, joka on predikaattisymbolin P tulkinta.
- Konnektiivien muodostettujen lauseiden totuus määräytyy kuten propositiologiikassa.
- Kvanttorit antavat mahdollisuuden eksistentiaaliseen ja universaaliin kvantifiointiin:
 - Lause $\exists x \phi(x)$ on tosi annetussa mallissa, jos lauseesta ϕ saadaan tosi sijoittamalla muuttujan x tulkinnaksi jokin olio.
 - Lause $\forall x \phi(x)$ on tosi annetussa mallissa, jos lause ϕ on tosi sijoitettiinpa muuttujan x tulkinnaksi mikä tahansa olio.

luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.4/12

- Sisäkkäisten kvanttorien kanssa täytyy olla huolellinen! Esim. kirjan sivulla 252 on virheitä.
- Kvanttorien suhde: Koska $\forall$ voidaan ajatella ikään kuin konjunktiksi yli maailman olioiden ja $\exists$ disjunktiksi, niin ne noudattavat de Morganin sääntöjä.
- Yhtäsuuruus on erikoisasemassa oleva relaatio, jonka tulkintaa ei voi asettaa. Termit t_1 ja t_2 ovat tässä relaatiossa vain jos niiden tulkinta on sama olio.

luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.5/12

ENSIMMÄISEN KERTALUVUN LOGIIKAN KÄYTTÖ

- Tietokantaan syötetään *aksioomat*, jotka ovat usein sovellusalueen käsitteiden määritelmiä, siis ekvivalenssilauseita jotka spesifioivat riittävät ja välttämättömät ehdot käsitteille (kuten $\text{person} \leftrightarrow \dots$).
- Voi olla myös sellaisia aksioomia, jotka eivät ole määritelmiä, vaan spesifioivat vain riittäviä ($\dots \rightarrow \text{person}$) tai välttämättömiä ($\text{person} \rightarrow \dots$) ehtoja.
- Lisäksi aksioomiksi otetaan *faktoja* jotka vain assertoivat tiettyjä tosiasioita (esim. $\text{person}(\text{John})$).
- Näistä aksioomista voidaan sitten johtaa *teoreemoja* (theorem). Teoreemoja ei tarvitse välttämättä tallettaa tietämyskantaan, koska ne eivät lisää päättelyvoimaa, mutta usein niitä talletetaan päättelyn nopeuttamiseksi.

luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.6/12

ESIMERKKI: SUKULAISUUSSUhteita

$$\forall m, c \text{ Äiti}(c) = m \\ \leftrightarrow \text{Nainen}(m) \wedge \text{Vanhempi}(m, c).$$

$$\forall w, h \text{ Aviomies}(h, w) \leftrightarrow \text{Mies}(h) \wedge \text{Puoliso}(h, w).$$

$$\forall x \text{ Mies}(x) \leftrightarrow \neg \text{Nainen}(x).$$

$$\forall p, c \text{ Vanhempi}(p, c) \leftrightarrow \text{Lapsi}(c, p).$$

$$\forall g, c \text{ Isovanhempi}(g, c) \\ \leftrightarrow \exists p \text{ Vanhempi}(g, p) \wedge \text{Vanhempi}(p, c).$$

$$\forall x, y \text{ Sisarus}(x, y) \\ \leftrightarrow x \neq y \wedge \exists p \text{ Vanhempi}(p, x) \wedge \text{Vanhempi}(p, y).$$

luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.7/12

ESIMERKKI: LUONNOLLISET LUVUT

- Luonnollisten lukujen teoria esitetään Peanon aksioomien muodossa. Oletetaan vakiosymboli 0 (tosin Peano itse taisi aloittaa luonnolliset luvut ykkösestä) ja funktiosymboli S.
- Luonnolliset luvut, merk. NatNum , määritellään seuraavasti:
 - $\text{NatNum}(0)$
 - $\forall n \text{ NatNum}(n) \rightarrow \text{NatNum}(S(n))$
 Siis 0 on luonnollinen luku ja jokaisen luonnollisen luvun seuraaja on luonnollinen luku. Seuraajafunktiota rajoitetaan seuraavasti:
 - $\forall n 0 \neq S(n)$
 - $\forall m, n m \neq n \rightarrow S(m) \neq S(n).$
- Nyt voidaan määritellä yhteenlasku seuraajafunktion avulla:
 - $\forall m \text{ NatNum}(m) \rightarrow +(0, m) = m$
 - $\forall m, n \text{ NatNum}(m) \wedge \text{NatNum}(n) \rightarrow +(S(m), n) = S(+(m, n)).$

luku08.tex – 58066-7 Tehtävyt (4 ov / 8 opt) – Rauli Hakli – 3/11/2005 – 13:33 – p.8/12

- Aksiooma palauttaa yhteenlaskun seuraajafunktion toistamiseen: Esimerkiksi $2 + 3$ esitetään muodossa $+(S(S(0)), S(S(S(0))))$ ja määritelmien perusteella saadaan $+(S(S(0)), S(S(S(0)))) = S(+(S(0), S(S(S(0)))))) = S(S(+(0, S(S(S(0)))))) = S(S(S(S(S(0))))))$, joka on siis 5.
- Prefix-muoto korvataan yleensä infix-muodolla: $+(0, m) \rightarrow 0 + m$ ja $S(n)$ muodolla $n + 1$.
- Vastaavasti voidaan määritellä kertolasku toistettuna yhteenlaskuna, potenssit toistettuna kertolaskuna, kokonaislukujen jakolasku, jakojäännös, alkuluvut jne joten koko aritmetiikka voidaan rakentaa käyttämällä yhtä vakiosymbolia, yhtä funktiosymbolia, yhtä predikaattisymbolia ja neljää aksioomaa.
- (Nyt ei oteta lukuun induktioperiaatetta, joka täytyisi muotoilla toisen kertaluvun logiikassa)

- Ympäristön ominaisuudet voidaan nyt kuvata selvästi helpommin kuin lauselogiikalla. Esim.
 $\forall x, y, a, b \text{ } Adjacent([x, y], [a, b]) \leftrightarrow [a, b] \in \{[x + 1, y], [x - 1, y], [x, y + 1], [x, y - 1]\}$
- Agentin sijainti voidaan ilmaista At -predikaatilla:
 $At(Agent, s, t)$, missä s on ruutu ja t on aika.
- Nyt agentin olisi hyödyllistä päätellä esim. seuraavasti:

$$\forall s, t \text{ } At(Agent, s, t) \wedge Breeze(t) \rightarrow Breezy(s).$$

- Diagnostisia sääntöjä:
 $\forall s \text{ } Breezy(s) \rightarrow \exists r \text{ } Adjacent(r, s) \wedge Pit(r)$
 $\forall s \neg Breezy(s) \rightarrow \neg \exists r \text{ } Adjacent(r, s) \wedge Pit(r)$
- Kausaalisääntöjä:
 $\forall r \text{ } Pit(r) \rightarrow [\forall s \text{ } Adjacent(r, s) \rightarrow Breezy(s)]$
 $\forall s [\forall r \text{ } Adjacent(r, s) \rightarrow \neg Pit(r)] \rightarrow \neg Breezy(s)$

- Havainnoissa havaintovektori ja ajanhetki, esim.
 $Percept([Stench, Breeze, Glitter, None, None], 5)$.
- Tekoja esitetään seuraavilla termeillä:
 $Turn(Right), Turn(Left), Forward, Shoot, Grab$
- Halutessaan tietää parhaan teon hetkellä 5 agentti tekee kyselyn $\exists a \text{ } BestAction(a, 5)$
- Vastaukseksi saadaan muuttujasidonta kuten $\{a/Grab\}$, jolloin agentti ilmoittaa tietämyskannalle tekevänsä teon Grab ja tekee sen.
- Havainnoista saadaan tietoja esim. seuraavasti:
 $\forall t, s, g, m, c \text{ } Percept([s, Breeze, g, m, c], t) \rightarrow Breeze(t)$
 $\forall t, s, b, m, c \text{ } Percept([s, b, Glitter, m, c], t) \rightarrow Glitter(t)$
- Heijastetointa voidaan toteuttaa säännöillä kuten
 $\forall t \text{ } Glitter(t) \rightarrow BestAction(Grab, t)$

TIETÄMYSINSINÖÖRITYÖN VAIHEET

Tietämyskannankonstruointiprosessi, jossa jonkin erikoisalueen tietämys koodataan tietämyskantaan, koostuu seuraavista vaiheista:

- Tehtävän identifiointi
- Relevantin tietämyksen keruu
- Terminologian määrittäminen: predikaattisymbolit, funktiosymbolit, vakiot
- Aihealueen tietämyksen koodaaminen
- Eryistysten ongelmatapausten kuvausten koodaaminen
- Kyselyiden muodostaminen tietämyskannalle
- Tietämyskannan testaaminen

Esimerkkinä piireihin liittyvän tietämyksen koodaaminen