F. Drewes 2004-09-24

Logik för datavetare, HT04

– Gruppövning 3 –

(1) Antag att ni vill beskriva blockvärldsexemplet predikatlogiskt. Inga konstanter motsvarande K_1, K_2, P_1, P_2 används och modeller får bestå av ett godtyckligt antal kuber och pyramider. Funktionssymboler finns inte och relationssymbolerna är de som har betraktats tidigare.

(a) Konstruera och diskutera några meningar som kan behövas i en formelmängd Φ vars syfte är att beskriva mängden av alla tillåtna världar (som på första gruppövningen fast nu inte med målet att skapa en fullständig beskrivning).

(b) Den typ av predikatlogik som har diskuterats tycks inte kunna beskriva egenskapen: ”Det kan inte finnas två objekt som ligger på samma objekt.“ Varför? Kan ni komma på någon utökning av predikatlogiken som klarar av det här?

(2) Låt $\mathbf{T} = (\mathbf{V}, \mathbf{K}, \mathbf{F})$ vara en signatur. Definiera induktivt utökningen av en substitution $\sigma\colon \mathbf{V} \to \mathsf{terms}(\mathbf{T})$ till en funktion $\sigma\colon \mathsf{terms}(\mathbf{T}) \to \mathsf{terms}(\mathbf{T})$, d.v.s. definiera $t\sigma$ för varje term $t \in \mathsf{terms}(\mathbf{T})$. (Kom ihåg den icke induktiva definitionen som gavs på föreläsningen.)

(3) Bevisa per induktion över termernas struktur att ekvationen $t(\sigma_1\sigma_2) = (t\sigma_1)\sigma_2$ som nämndes på föreläsningen är korrekt.

(4) Varför är substitutionen $\sigma_1 = \{f(y)/x\}$ inte generellare än $\sigma_2\{f(a)/x\}$ (där x, y är variabler och a är en konstant), d.v.s. varför gäller inte $\sigma_2 = \sigma_1\sigma$ där $\sigma = \{a/y\}$? (Applicera både σ_2 och $\sigma_1\sigma$ på $P(x, y)$ och förklara resultaten.)

(5) Hitta mgu:n för följande par av atomära formler om den finns (där w, x, y, z är variablerna):

$$\begin{array}{rcl} P(x, f(y)) & \text{och} & P(f(y), f(x)) \\ P(x, g(y, x)) & \text{och} & P(f(y), g(f(z), f(f(w)))) \\ P(x, g(y, x)) & \text{och} & P(f(y), g(f(a), f(g(z, z)))). \end{array}$$

Kommentarer…

(1) (a) En mening är t.ex. $(\forall x)(\mathsf{Is\_cube}(x) \leftrightarrow \neg\mathsf{Is\_pyramid}(x))$ – varje objekt är antingen en kub eller en pyramid. En annan är $(\forall x)(\mathsf{Is\_pyramid}(x) \rightarrow (\forall y)(\neg\mathsf{On}(y,x)))$ – inget objekt kan ligga på en pyramid. En tredje mening som kan diskuteras är $(\forall x)(\mathsf{On\_table}(x) \leftrightarrow (\neg(\exists y)(\mathsf{On}(x,y))))$ – varje objekt ligger antingen på bordet eller på en annan. En fjärde är $\neg(\exists x)(\mathsf{On}(x,x))$ – inget objekt ligger på sig själv.

(b) Man skulle behöva möjligheten att t.ex. skriva

$$(\forall x)(\forall y)(\forall z)((\mathsf{On}(x,z) \land \mathsf{On}(y,z)) \rightarrow x = y)$$

dvs man behöver en *predikatlogik med likhet*, alltså med en speciell binär relationssymbol '=' **som alltid tolkas som likhet**.

Viktigt: Det räcker inte att lägga till en relationssymbol $\mathsf{EQUAL}^{(2)}$ eftersom predikatlogiken inte tillåter att uttrycka att den alltid ska tolkas som likhet. En dellösning vore att i Φ lägga till axiomer som uttrycker att EQUAL är reflexiv, transitiv och symmetriskt, samt att alla relationssymboler $P^{(n)}$ uppfyller

$$(\forall x_1)(\forall y_1)\cdots(\forall x_n)(\forall y_n)((\mathsf{EQUAL}(x_1,y_1) \land \cdots \land \mathsf{EQUAL}(x_n,y_n)) \rightarrow (P(x_1,\ldots,x_n) \leftrightarrow P(y_1,\ldots,y_n))).$$

och alla funktionssymboler $f^{(n)}$

$$(\forall x_1)(\forall y_1)\cdots(\forall x_n)(\forall y_n)((\mathsf{EQUAL}(x_1,y_1) \land \cdots \land \mathsf{EQUAL}(x_n,y_n)) \rightarrow \mathsf{EQUAL}(f(x_1,\ldots,x_n), f(y_1,\ldots,y_n)))$$

men det är krångligt, är inte exakt det man av praktiska skäl vill ha (eftersom det ger en godtycklig ekvivalensrelation istället för en "riktig" likhet) och producerar oändligt många formler när man har oändligt många relations- eller funktionssymboler.

(2) För varje konstant $a \in \mathbf{K}$ definieras $a\sigma = a$ och för varje term $t = f(t_1,\ldots,t_n)$ där $f^{(n)} \in \mathbf{F}$ och $t_1,\ldots,t_n \in \mathsf{terms}(\mathbf{T})$

$$t\sigma = f(t_1\sigma,\ldots,t_n\sigma).$$

(För variabler x är $x\sigma$ definierad enligt antagande.)

(3) *Induktionsbas:* För variabler x har kompositionen $\sigma_1\sigma_2$ definierats som $x(\sigma_1\sigma_2) = (x\sigma_1)\sigma_2$. För $a \in \mathbf{K}$ är enligt definitionen ovan $a(\sigma_1\sigma_2) = a = (a\sigma_1) = (a\sigma_1)\sigma_2$.

Induktionsantagande: Påståendet gäller för termer $t_1,\ldots,t_n \in \mathsf{terms}(\mathbf{T})$.

Induktionssteg: För varje term $t = f(t_1,\ldots,t_n)$ gäller

$$\begin{array}{rcll} t(\sigma_1\sigma_2) & = & f(t_1(\sigma_1\sigma_2),\ldots,t_n(\sigma_1\sigma_2)) & \text{(definition i (2))} \\ & = & f((t_1\sigma_1)\sigma_2,\ldots,(t_n\sigma_1)\sigma_2)) & \text{(induktionsantagande)} \\ & = & f(t_1\sigma_1,\ldots,t_n\sigma_1)\sigma_2 & \text{(definition i (2))} \\ & = & (t\sigma_1)\sigma_2 & \text{(definition i (2)).} \end{array}$$

(4) $\sigma_1\sigma$ är lika med $\{f(a)/x, a/y\}$ vilket inte är σ_2.

(5) Robinsons algoritm ger

φ	φ'	σ_{mgu}
$P(x, f(y))$	$P(f(y), f(x))$	
$P(f(y), f(y))$	$P(f(y), f(f(y)))$	$f(y)/x$
Avbrott - $f(y)$ innehåller y		

φ	φ'	σ_{mgu}
$P(x, g(y, x))$	$P(f(y), g(f(z), f(f(w))))$	
$P(f(y), g(y, f(y)))$	$P(f(y), g(f(z), f(f(w))))$	$f(y)/x$
$P(f(f(z)), g(f(z), f(f(z))))$	$P(f(f(z)), g(f(z), f(f(w))))$	$f(f(z))/x$, $f(z)/y$
$P(f(f(w)), g(f(w), f(f(w))))$	$P(f(f(w)), g(f(w), f(f(w))))$	$f(f(w))/x$, $f(w)/y$, w/z
Klar (alternativt $\{f(f(z))/x, f(z)/y, z/w\}$ i sista steget)		

φ	φ'	σ_{mgu}
$P(x, g(y, x))$	$P(f(y), g(f(a), f(g(z, z))))$	
$P(f(y), g(y, f(y)))$	$P(f(y), g(f(a), f(g(z, z))))$	$f(y)/x$
$P(f(f(a)), g(f(a), f(f(a))))$	$P(f(f(a)), g(f(a), f(g(z, z))))$	$f(f(a))/x, f(a)/y$
Avbrott - varken $f(a)$ eller $g(z, z)$ är en variabel		