Ludwig-Maximilians-Universität München **SoSe 2014**
Institut für Informatik
Dr. M. Hölzl, C. Kroiß

Formale Techniken der Software-Entwicklung
Übungsblatt 4
Besprechung am 23.05.2014

Musterlösung

Die Aufgaben 1 und 2 wurden aus [1] übernommen und sind daher auf Englisch.

Aufgabe 1:

Let $F(x, y)$ mean that x is the father of y; $M(x, y)$ denotes x is the mother of y. Similarly, $H(x, y)$, $S(x, y)$, and $B(x, y)$ say that x is the husband/sister/brother of y, respectively. You may also use constants to denote individuals, like 'Ed' and 'Patsy.' However, you are not allowed to use any predicate symbols other than the above to translate the following sentences into predicate logic:

(a) Everybody has a mother.

Lösung:
$\forall x.\exists y.M(y, x)$

(b) Everybody has a father and a mother.

Lösung:
$\forall x.\exists y.\exists z.M(y, x) \land F(z, x)$

(c) Whoever has a mother has a father.

Lösung:
$\forall x.(\exists y.M(y, x)) \to (\exists z.F(z, x))$

(d) Ed is a grandfather.

Lösung:
$\exists x.\exists y.F(Ed, x) \land (F(x, y) \lor M(x, y))$

(e) All fathers are parents.

Lösung:
$\forall x.(\exists y.F(x,y)) \rightarrow (\exists z.F(x,z) \vee M(x,z))$

(f) All husbands are spouses.

Lösung:
Da es kein Prädikat für "wife" gibt, ist irgendwie nicht klar, wie man "spouse" anders als durch die "husband"-Beziehung ausdrücken soll. Man könnte hier noch Monogamie fordern:
$\forall x.\forall y.H(x,y) \rightarrow (\forall z.H(z,y) \rightarrow z = x)$

(g) No uncle is an aunt.

Lösung:
$$\forall x.(\exists y.\exists z.B(x,y) \wedge (M(y,z) \vee F(y,z))) \rightarrow \neg(\exists y.\exists z.S(x,y) \wedge (M(y,z) \vee F(y,z)))$$

(h) All brothers are siblings.

Lösung:
$\forall x.\forall y.B(x,y) \rightarrow (B(y,x) \vee S(y,x))$

(i) Nobody's grandmother is anybody's father.

Lösung:
$\forall x.(\exists y.\exists z.M(x,y) \wedge (M(y,z) \vee F(y,z))) \implies \neg(\exists y.F(x,y))$

(j) Ed and Patsy are husband and wife.

Lösung:
Wie oben kann man hier Monogamie fordern:
$H(Ed,Patsy) \wedge (\forall x.H(Ed,x) \rightarrow x = Patsy) \wedge (\forall y.H(y,Patsy) \rightarrow y = Ed)$

(k) Carl is Monique's brother-in-law.

Lösung:
$(\exists x.B(Carl,x) \wedge H(x,Monique)) \vee (\exists y.H(Carl,y) \wedge S(Monique,y))$

Aufgabe 2:
The following sentences are taken from the RFC3157 Internet Taskforce Document 'Securely Available Credentials – Requirements.' Specify each sentence in predicate logic, defining predicate symbols as appropriate:

Lösung:

$loggedIn(a,s)$ bla

(a) An attacker can persuade a server that a successful login has occurred, even if it hasn't.

Lösung:
Version 1:

$$\exists a \exists s. \neg loggedIn(a, s) \rightarrow canPersuade(a, s)$$

Version 2: Let σ denote a *state*.

$$\exists a \exists s \exists \sigma. possible(\sigma) \wedge (\neg loggedIn(a, s, \sigma) \rightarrow accessGranted(s, a, \sigma))$$

(b) An attacker can overwrite someone else's credentials on the server.

Lösung:

$$\exists u \exists c \exists s \exists d. \neg ownsCredentials(u, c) \rightarrow canWrite(u, c, s, d)$$

(c) All users enter passwords instead of names.

Lösung:

$$\forall u \forall s. attemptLogin(u, s) \rightarrow (enterPassword(u, s) \wedge \neg enterName(u, s)).$$

(d) Credential transfer both to and from a device MUST be supported.

Lösung:

$$\forall u \forall d \forall s \forall c. (uses(u, d) \wedge connected(d, s) \wedge ownsCredentials(u, c)) \rightarrow (canWrite(u, c, s, d) \wedge canRead(u, c, s, d))$$

(e) Credentials MUST NOT be forced by the protocol to be present in cleartext at any device other than the end user's.

Lösung:

$$\forall d \forall c \forall u. \neg ownsDevice(u, d) \rightarrow \neg forcedCleartext(c, d)$$

Version 2:

$$\exists \sigma \exists d \exists c \exists u. possible(\sigma) \wedge \neg ownsDevice(u, d, \sigma) \wedge \neg cleartext(c, d, \sigma)$$

(f) The protocol MUST support a range of cryptographic algorithms, including symmetric and asymmetric algorithms, hash algorithms, and MAC algorithms.

Lösung:

$$\begin{aligned} \exists a_1 \exists a_2 \exists a_3 \exists a_4. & symmetricAlgorithm(a_1) \wedge supported(a_1) \wedge \\ & asymmetricAlgorithm(a_2) \wedge supported(a_2) \wedge \\ & hashAlgorithm(a_3) \wedge supported(a_3) \wedge \\ & macAlgorithm(a_4) \wedge supported(a_4) \end{aligned}$$

(g) Credentials MUST only be downloadable following user authentication or else only downloadable in a format that requires completion of user authentication for deciphering.

Lösung:
$\forall u \forall c \forall s.(canRead(u,c,s) \rightarrow loggedIn(u,s)) \wedge (\neg loggedIn(u,s) \rightarrow canReadEncrypted(u,c,s))$

(h) Different end user devices MAY be used to download, upload, or manage the same set of credentials.

Lösung:
Definiere
$canHandle(x,c) \equiv canDownload(x,c) \wedge canUpload(x,c) \wedge canManage(x,c)$.
Dann ist eine direkte Übersetzung der Aussage einfach

$$\begin{aligned}\Phi &\equiv \forall x.\forall y.\forall c.(canHandle(x,c) \wedge canHandle(y,c)) \rightarrow (x = y \vee x \neq y) \\ &\equiv \forall y.\forall c.(canHandle(x,c) \wedge canHandle(y,c)) \rightarrow \top\end{aligned}$$

Aufgabe 3:

Gegeben sei die Formel $\Phi \stackrel{def}{=} \forall x \forall y.Q(g(x,y),g(y,y),z)$. Finden Sie zwei Modelle $\mathcal{M}$ und $\mathcal{M}'$ so dass $\mathcal{M} \models \Phi$ aber $\mathcal{M}' \not\models \Phi$ gilt.

Lösung:
Hier gibt es natürlich unendlich viele verschiedene Möglichkeiten. Eine ganz triviale folgt:
Gegeben eine Struktur $\mathcal{A}$ mit Trägermenge $\mathbb{N}$ und

$$g^{\mathcal{A}}(x,y) = 1 \quad \forall x,y \in \mathbb{N}$$
$$Q^{\mathcal{A}} = \{(1,1,1)\}$$

.
Dann wähle für $\mathcal{M} = (\mathcal{A}, w)$:

$$w(z) = 1$$

und für $\mathcal{M}' = (\mathcal{A}, w')$:

$$w(z) = 0$$

Es ist leicht zu sehen, dass $\mathcal{M} \models \Phi$ aber $\mathcal{M}' \not\models \Phi$ gilt.

Aufgabe 4:

Gegeben sei der folgende Satz:

$$\Phi \stackrel{def}{=} \forall x \exists y \exists z(P(x,y) \wedge P(z,y) \wedge (P(x,z) \rightarrow P(z,x)))$$

Welche der folgenden Modelle erfüllen Φ?

(a) Das Modell $\mathcal{M}$ besteht aus den natürlichen Zahlen mit $P^{\mathcal{M}} \stackrel{def}{=} \{(m,n)|m < n\}$.

Lösung:
Gilt.
Wähle $y > x$ und $z = x$. Dann gilt $\neg P(x, z)$ und somit ist Φ wahr.

(b) Das Modell $\mathcal{M}'$ besteht aus den natürlichen Zahlen mit $P^{\mathcal{M}'} \stackrel{def}{=} \{(m, 2 * m) | m \in \mathbb{N}\}$.

Lösung:
Gilt.
Wähle $y = 2 * x$ und $z = x$. Wenn man 0 zu den natürlichen Zahlen rechnet, dann gilt im Fall $x = 0$ auch $P(x, z) = P(0, 0) = (0, 2 * 0) \in P^{\mathcal{M}}$ und genauso auch $P(z, x)$. Somit gilt die Implikation in der Klammer. Für alle anderen Fälle $x > 0$ ist bei $z = x$ die Prämisse der Implikation falsch und somit die Implikation in der Klammer wahr.

(c) Das Modell $\mathcal{M}''$ besteht aus den natürlichen Zahlen mit $P^{\mathcal{M}''} \stackrel{def}{=} \{(m, n) | m < n + 1\}$.

Lösung:
Gilt.
Wähle $z = x$ und $y \geq x$. Dann gilt $P(x, y) \wedge P(z, y) \wedge P(x, z) \wedge P(z, x)$ und somit ist Φ wahr.

Aufgabe 5:
Geben Sie Beweisbäume im Sequenzkalkül für folgende Formeln an:

(a) $\exists x(P \to Q(x)) \to (P \to \exists z.Q(z))$, wobei P ein aussagenlogisches Symbol und Q ein unäres Predikat ist.

Lösung:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P \vdash P, Q(t)} \quad \dfrac{}{Q(t), P \vdash Q(t)}}{P \to Q(t), P \vdash Q(t)}}{P \to Q(t), P \vdash \exists z.Q(z)}}{P \to Q(t) \vdash P \to \exists z.Q(z)}}{\vdash (P \to Q(t)) \to (P \to \exists z.Q(z))}}{\vdash \exists x.(P \to Q(x)) \to (P \to \exists z.Q(z))}$$

(b) $(\forall x.A) \vee (\forall x.B) \to \forall x(A \vee B)$, wobei A und B beliebige Formeln seien, in denen x nicht frei vorkommt.

Lösung:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A[y/x] \vdash A[y/x], B[y/x]}}{\forall x.A \vdash A[y/x], B[y/x]} \quad \dfrac{\dfrac{}{B[y/x] \vdash A[y/x], B[y/x]}}{\forall x.B \vdash A[y/x], B[y/x]}}{(\forall x.A) \vee (\forall x.B) \vdash A[y/x], B[y/x]}}{(\forall x.A) \vee (\forall x.B) \vdash \forall x.(A \vee B)}}{\vdash (\forall x.A) \vee (\forall x.B) \to \forall x.(A \vee B)}$$

Literatur

[1] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and reasoning about systems.* Cambridge University Press, 2004.