

Intelligente Systeme im WWW

Dr. Pascal Hitzler, Dr. York Sure, Markus Krötzsch, Daniel Poltze
Sommersemester 2005

Lösung zu Übung 4 (20.6.2005)

Aufgabe 4.2 Wandeln Sie die beiden Wissensbasen aus Aufgabe 4.1 in Klauselform um. Drücken Sie dazu die Formeln zunächst in FOL aus und wandeln Sie diese anschließend in Klauseln um.

Man kann diese Umwandlungen auch gemäß der oben angegebenen intuitiven Bedeutung „im Kopf“ durchführen. Bei Betrachtung des Ergebnisses sollte die inhaltliche Gleichwertigkeit sichtbar sein.

$\text{Pizza} \sqcap \text{PizzaBelag} \sqsubseteq \perp$	$(\neg \text{Pizza}(X) \vee \neg \text{PizzaBelag}(X))$
$\exists \text{ hatBelag.PizzaBelag} \sqsubseteq \text{Pizza}$	$(\neg \text{hatBelag}(X, Y) \vee \neg \text{PizzaBelag}(Y) \vee \text{Pizza}(X))$
$\text{Student} \sqsubseteq \exists \text{ besucht.Vorlesung}$	$(\neg \text{Student}(X) \vee \text{besucht}(X, f(X))) \wedge$ $(\neg \text{Student}(X) \vee \text{Vorlesung}(f(X)))$
$\text{Vorlesung} \sqsubseteq \exists \text{ besuchtVon.}(\text{Student} \sqcap \text{Fleißig})$	$(\neg \text{Vorlesung}(X) \vee \text{besuchtVon}(X, g(X))) \wedge$ $(\neg \text{Vorlesung}(X) \vee \text{Student}(g(X))) \wedge$ $(\neg \text{Vorlesung}(X) \vee \text{Fleißig}(g(X)))$

Die nicht genannten (ABox-)Axiome sind bereits in Klauselform. Im zweiten Teil sind f und g neue Skolem-Funktionen, die zur Auflösung der Existenzquantoren eingeführt wurden.

Aufgabe 4.3 Modellieren Sie die folgenden Zusammenhänge aus der griechischen Mythologie mit Hilfe der Ontologiesprache F-Logik. Verwenden Sie *Nesting* (Einbettung) zur Vereinfachung der Darstellung.

Zeus, Sohn des Gottes Kronos und der Göttin Rhea, ist der oberste olympische Gott in der griechischen Mythologie. Götter und Göttinnen sind Unsterbliche, während Männer und Frauen zu den Sterblichen zählen. Zeus' erste Gattin war Metis, ihr folgten Themis und schließlich Hera, deren Eltern ebenfalls Kronos und Rhea waren. Alkmene, die sterbliche Gattin des Mannes Amphitryon, hat gemeinsam mit Zeus den Sohn Herakles.

Die Hydra ist ein Seeungeheuer mit mindestens neun verschiedenen Köpfen, welches durch Herakles bekämpft wurde. Seeungeheuer sind spezielle Ungeheuer. Ein Sterblicher, der gegen ein Ungeheuer kämpft, ist ein Held.

Die direkten Vorfahren einer Person sind ihr Vater und ihre Mutter. Ein Halbgott ist eine Person, die sowohl Menschen als auch Götter unter ihren direkten Vorfahren hat. Halbgötter sind sterblich.

(Weitere Details sind in der Wikipedia zu finden.)

```

göttin::unsterblich.
gott::unsterblich.
frau::sterblich.
mann::sterblich.
seeungeheuer::ungeheuer.
zeus:gott[vater->kronos:gott; mutter->rhea:göttin;
          gattin@(1)->metis; gattin@(2)->themis; gattin@(3)->hera].
hera:göttin[vater->kronos; mutter->rhea].
alkmene:frau[gatte->>amphitryon].
herakles:mann[mutter->alkmene; vater->zeus].
hydra:seeungeheuer[kopf->>{kopf1,kopf2,kopf3,kopf4,kopf5,kopf6,
                          kopf7,kopf8,kopf9}].
bekämpft(herakles,hydra).
FORALL X held(X) <- X:sterblich AND EXISTS Y (Y:ungeheuer AND
                                             bekämpft(X,Y)).
FORALL X,Y dirVor(X,Y) <- X[vater->Y] OR X[mutter->Y].
FORALL X halbgott(X) <- EXISTS Y (dirVor(X,Y) AND Y:sterblich) AND
                               EXISTS Z (dirVor(X,Z) AND Z:unsterblich).
FORALL X X:sterblich <- halbgott(X).

```

Anmerkung: Die Umschreibung der Neunköpfigkeit der Hydra mit Hilfe neuer Individuen ist ein Notbehelf – Zahlen sind in F-Logik ansonsten nur mit einem (hier nicht behandeltem) Built-In verwendbar.

Aufgabe 4.4 Modellieren Sie die Wissensbasis aus Aufgabe 4.3 mit den Ausdrucksmitteln der Beschreibungslogik.

ABox:

Gott(zeus)	Gott(kronos)	vater(zeus, kronos)	mutter(zeus, rhea)
gattin(zeus, metis)	gattin(zeus, themis)	gattin(zeus, hera)	Göttin(rhea)
Göttin(hera)	gattin(zeus, hera)	vater(hera, kronos)	mutter(hera, rhea)
Frau(alkmene)	Mann(amphitryon)	Mann(herakles)	vater(herakles, zeus)
gattin(amphytrion, alkmene)		mutter(herakles, alkmene)	
{hydra} \sqsubseteq Seeungeheuer \sqcap ≥ 9 hatKopf		bekämpft(herakles, hydra)	

TBox:

Göttin \sqsubseteq Unsterblich	Gott \sqsubseteq Unsterblich	Frau \sqsubseteq Sterblich
Mann \sqsubseteq Sterblich	Seeungeheuer \sqsubseteq Ungeheuer	Halbgott \sqsubseteq Sterblich
Held \sqsupseteq Sterblich \sqcap \exists bekämpft.Ungeheuer		
Halbgott \equiv \exists dirvor.Sterblich \sqcap \exists dirvor.Unsterblich		

RBox:

vater \sqsubseteq dirvor	mutter \sqsubseteq dirvor
----------------------------	-----------------------------

Aufgabe 4.5 Betrachten Sie die folgenden Erweiterungen der obigen Wissensbasis:

Nur ein schlechter Sohn würde gegen seinen eigenen Vater kämpfen. Kronos aber hatte seinen eigenen Vater bekämpft und entmachtet, und auch Zeus stürzte Kronos später im Kampf. Herakles dagegen hat seinen Vater nicht bekämpft.

Zu den Vorfahren einer Person zählen nicht nur ihre Vorfahren, sondern auch alle direkten Vorfahren ihrer Vorfahren.¹

- (a) Erweitern Sie Ihre F-Logik-Spezifikation aus Aufgabe 4.3 um diese zusätzlichen Informationen.

Die in der Übung vorgestellte Lösung enthielt leider einen Fehler: EXISTS darf in Regeln der F-Logik nur im Rumpf auftauchen. FORALL darf dagegen auch in Rümpfen vorkommen, da es dort (wie im Beispiel in der Vorlesung) durch einen negierten Kopf einer Hilfregel ersetzt werden kann.

Die Existenz eines nicht bekannten Vaters kann allerdings nicht ausgedrückt werden. Deshalb sei für den F-Logik-Teil der Aufgabe zusätzlich gegeben, dass der Vater von Kronos Uranos war.

```
kronos[vater->uranos] AND bekämpft(kronos,uranos).
bekämpft(zeus,kronos).
NOT bekämpft(herakles,zeus).
```

```
FORALL X badSon(X) <- EXISTS Y (X[vater->Y] AND bekämpft(X,Y)).
FORALL X,Y vorfahre(X,Y) <- dirVor(X,Y).
FORALL X,Y vorfahre(X,Y) <- EXISTS Z vorfahre(X,Z) AND dirVor(Z,Y).
```

- (b) Können die obigen Definitionen von „Vorfahre“ und „schlechter Sohn“ jeweils ohne Regelerweiterungen in DL beschrieben werden? Wie ist es mit dem Fakt, dass Kronos seinen Vater bekämpft hat?

„schlechter Sohn“ kann nicht beschrieben werden. Die anderen Aussagen werden wie folgt formuliert:

```
dirvor  $\sqsubseteq$  vorfahre
transitiv(vorfahre)
 $\exists$  vater-.{kronos}  $\sqsubseteq$   $\exists$  bekämpft-.{kronos}
```

Man verdeutliche sich dabei besonders die Idee für die Modellierung des unbekanntem Vaters von Kronos.

¹Offensichtlich ist der Begriff „Vorfahren“ also *rekursiv* definiert.

- (c) Erweitern Sie nun, soweit möglich, auch Ihre DL-Spezifikation aus Aufgabe 4.4 um die neuen Fakten. Beschreiben Sie die nicht in DL darstellbaren Begriffe durch SWRL-Regeln.

Die zusätzlichen DL-Axiome sind in der vorigen Teilaufgabe angegeben. Die zusätzliche Regel lautet:

$\text{vater}(?x, ?y) \wedge \text{bekämpft}(?x, ?y) \Rightarrow \text{BadSon}(?x)$

- (d) Geben Sie zu jeder unter (c) eingeführten Regel eine Variante an, die DL-safe ist. Führen Sie dazu ein Hilfsprädikat $O(X)$ für alle bekannten Individuen ein, wie in der Vorlesung besprochen.

$\text{BadSon}(X) \leftarrow \text{vater}(X, Y), \text{bekämpft}(X, Y), O(X), O(Y)$

Außerdem wird für jedes bekannte Individuum i ein Fakt $O(i)$ angelegt.

- (e) Formulieren Sie in F-Logik Anfragen nach

- allen schlechten Söhnen, die Götter sind oder mit einer Göttin vermählt waren, und
- nach allen Sterblichen, die keine schlechten Söhne sind.

Welche Ergebnisse würde man jeweils erhalten?

FORALL X,Y <- badSon(X) AND (X:gott OR X[gattin->Y:göttin]).
{kronos, zeus}

FORALL X <- X:mann AND NOT badSon(X). {herakles, amphitryon}

Für die gegebenen Antworten spielt die Closed-World-Semantik eine wichtige Rolle, da zum Beispiel nichts über Amphitryons Vater bekannt ist.

- (f) Formulieren Sie die obigen Anfragen als DL-Klassen. Welche Individuen der erweiterten DL-Ontologie würden diesen Klassen jeweils zugeordnet? Wie verändert sich das Ergebnis, wenn man alle Regeln durch ihre DL-safe Variante ersetzt?

Klasse	SWRL	DL-safe
BadSon \sqcap (Gott \sqcup \exists gattin.Göttin)	{kronos, zeus}	{zeus}
\neg BadSon \sqcap Mann	{}	{}

Die zweite Anfrage produziert in keiner Variante ein Ergebnis, weil die Wissensbasis nicht ausschließt, dass eine Person noch weitere nicht genannte Väter hat, die sie bekämpft hat. Fügt man der Wissensbasis die Funktionalität der Vater-Beziehung hinzu:

$$\leq 1 \text{ vater} \sqsupseteq \top,$$

dann besteht immernoch die Möglichkeit, dass man aus anderen Gründen ebenfalls ein schlechter Sohn ist. Man müsste also noch Regeln anfügen, die ausdrücken, dass jeder schlechte Sohn seinen Vater bekämpft, damit man für irgendein Individuum mit Sicherheit \neg BadSon schlussfolgern kann.

Aufgabe 4.6 Wie in der Vorlesung besprochen, ist DLP der Teil von OWL-DL, den man als ein definites Logikprogramm gleicher Semantik ausdrücken kann.

- (a) Zeigen Sie, dass die folgenden Ausdrücke zu DLP gehören, indem Sie sie in semantisch äquivalente Horn-Klauseln umformen.

DL-Axiom	DLP-Regel(n)
$\text{Frau} \sqcup \text{Mann} \sqsubseteq \text{Mensch}$	$\text{Mensch}(X) \leftarrow \text{Frau}(X)$ $\text{Mensch}(X) \leftarrow \text{Mann}(X)$
$\text{elterVon} \equiv \text{kindVon}^{-}$	$\text{elterVon}(Y, X) \leftarrow \text{kindVon}(X, Y)$ $\text{kindVon}(Y, X) \leftarrow \text{elterVon}(X, Y)$
$\text{Mann} \sqcap \exists \text{kindVon}^{-} . \top \sqsubseteq \text{Vater}$	$\text{Vater}(X) \leftarrow \text{Mann}(X) \wedge \text{kindVon}(Y, X)$
$\text{elterVon} \sqsubseteq \text{vorfahreVon}$	$\text{vorfahreVon}(X, Y) \leftarrow \text{elterVon}(X, Y)$
$\text{Frau} \sqcap \exists \text{vorfahreVon} . \top \sqsubseteq \text{Mutter}$	$\text{Mutter}(X) \leftarrow \text{Frau}(X) \wedge \text{vorfahreVon}(Y, X)$
$\text{Mensch} \sqsubseteq \forall \text{vorfahreVon} . \text{Mensch}$	$\text{Mensch}(Y) \leftarrow \text{Mensch}(X) \wedge \text{vorfahreVon}(X, Y)$
$\{\text{marc}, \text{heiko}, \text{johanna}\} \sqsubseteq \text{Mensch}$	$\text{Mensch}(\text{marc})$ $\text{Mensch}(\text{heiko})$ $\text{Mensch}(\text{johanna})$

- (b) Die folgende Wissensbasis ist nicht in DLP. Wie könnte man das nachweisen?

$\{\text{gerrit}\} \sqsubseteq \text{Mensch}$
 $\text{Frau} \sqcup \text{Mann} \equiv \text{Mensch}$

Offensichtlich gibt es zwei mögliche minimale Modelle für die Wissensbasis: $\{\text{Mensch}(\text{gerrit}), \text{Frau}(\text{gerrit})\}$ und $\{\text{Mensch}(\text{gerrit}), \text{Mann}(\text{gerrit})\}$. Ein DLP-Programm ist aber ein (sehr einfaches) definites Logikprogramm und muss also ein kleinstes Modell haben. Da die obige Spezifikation kein kleinstes Modell zulässt, kann eine Übersetzung in DLP nicht möglich sein. Wir sehen daran, dass der Nachweis nicht in DLP zu sein im Allgemeinen nicht so einfach ist.

- (c) Ist die folgende Wissensbasis vollständig in DLP? Geben Sie gegebenenfalls eine (teilweise) Übersetzung in Horn-Logik an.

$\text{Eis} \sqcup \text{Früchte} \sqsubseteq \text{Süßspeise}$
 $\text{Nachtisch} \sqsubseteq \text{Eis} \sqcup \text{Früchte} \sqcup \text{Süßspeise}$
 $\top \sqsubseteq \forall \text{hatBelag}^{-} . \text{Pizza}$
 $\exists \text{hatBelag} . \top \sqcap \forall \text{hatBelag} . \perp \sqsubseteq \text{LeerePizza} \sqcup \text{PizzaCarnivorus}$

Die Wissensbasis ist (trotz vieler problematischer Ausdrücke) in DLP, weil sie semantisch zu folgender Wissensbasis äquivalent ist:

$\text{Eis} \sqcup \text{Früchte} \sqsubseteq \text{Süßspeise}$
 $\text{Nachtisch} \sqsubseteq \text{Süßspeise}$
 $\top \sqsubseteq \forall \text{hatBelag}^{-} . \text{Pizza}$
 $\perp \sqsubseteq \text{LeerePizza} \sqcup \text{PizzaCarnivorus}$

Das lässt sich nun leicht übersetzen:

Süßspeise(X) \leftarrow Eis(X) Süßspeise(X) \leftarrow Früchte(X)

Süßspeise(X) \leftarrow Nachtisch(X)

Pizza(X) \leftarrow hatBelag(X, Y)

Das letzte Axiom sagt nichts aus und fällt daher weg.

Wir sehen daran, dass der Nachweis in DLP zu sein im Allgemeinen nicht so einfach ist.