Technische Universität München
Informatik / Lehrstuhl 7
Esparza/Schwoon

WS 2007/08
Übungsblatt 13
25. Januar 2008

Übung zu Logik

Bearbeiten Sie diese Aufgaben bis zur nächsten Übung am Freitag, 1. Februar, um 11:50 Uhr. Die Lösungen werden in der Übung besprochen.

Aufgabe 1 Modellieren mit Prolog

Eine bekannte Denkaufgabe ist wie folgt:

> Am linken Ufer eines Flusses steht ein Mann, der einen Wolf, eine Ziege und einen Kohlkopf mit sich führt. Am Fluss ist ein kleines Boot, mit dem der Mann zwischen den Ufern hin- und herfahren kann. Das Boot ist so klein, dass außer dem Mann höchstens ein weiteres Objekt Platz hat. Der Mann will nun sich und den Wolf, die Ziege und den Kohlkopf ans rechte Ufer bringen.
>
> Lässt der Mann den Wolf und die Ziege unbeaufsichtigt an einem Ufer zurück, so wird der Wolf die Ziege fressen. Lässt der Mann die Ziege und den Kohlkopf allein zurück, so wird die Ziege den Kohlkopf fressen. Finden Sie eine Möglichkeit, wie der Mann alle drei heil ans rechte Ufer bringt.

Wir versuchen im Folgenden, das Rätsel zu lösen, indem wir `otter` als Prolog-Interpreter benutzen. Dazu stellen wir eine konkrete Situation jeweils als 4-Tupel dar, z.B. drücken wir mit (l, r, l, r) aus, dass der Mann am linken Ufer steht, der Wolf am rechten Ufer, die Ziege am linken Ufer und der Kohlkopf am rechten Ufer.

(a) Erstellen Sie Prozedur- und/oder Tatsachenklauseln, so dass $wolf_ok(M, W, Z, K)$ genau dann abgeleitet werden kann, wenn $(M, W, Z, K) \in \{l, r\}^4$ eine Situation darstellt, in der Wolf und Ziege nicht unbeaufsichtigt am selben Ufer sind. Stellen Sie in ähnlicher Weise Klauseln für *ziege_ok* auf, um auszudrücken, dass Ziege und Kohlkopf nicht unbeaufsichtigt am selben Ufer sind. Mit der Prozedurklausel

$$(wolf_ok(M, W, Z, K) \wedge ziege_ok(M, W, Z, K)) \rightarrow ok(M, W, Z, K)$$

erhalten wir dann eine Regel, die uns sagt, ob eine Situation "ungefährlich" ist.

(b) Mit dem Prädikat *erreichbar* wollen wir nun ausdrücken, dass (durch beliebiges Hin- und Herfahren mit dem Boot) eine Situation eintreten kann, ohne dass Ziege oder Kohlkopf gefährdet werden. Als Tatsachenklausel stellen wir auf, dass anfänglich alle vier auf dem linken Ufer sind:

$$erreichbar(l, l, l, l)$$

Erstellen Sie Prozedurklauseln, die die übrigen Situationen herleiten, die ohne Gefährdung von Ziege oder Kohlkopf erreichbar sind.

(c) Nehmen wir alle bisher aufgestellten Klauseln und erweitern sie um eine Zielklausel, die das Erreichen des rechten Ufers ausdrückt:

$$\neg erreichbar(r, r, r, r)$$

Wenn es eine Lösung gibt, so ist die Klauselmenge zusammen mit der Zielklausel unerfüllbar. Benutzen Sie `otter` zunächst wie gehabt (siehe das Informationsblatt zu `otter`), um herauszufinden, ob dies der Fall ist.

Der Resolutionsalgorithmus von `otter` arbeitet mit der Stützmengen-Restriktion. Dabei steht die erfüllbare Teilmenge von Formeln zwischen `list(usable)` und `end_of_list`, die übrigen wie gehabt stehen zwischen `list(sos)` und `end_of_list`. Nutzen Sie diese Restriktion nun aus, indem Sie alle Klauseln bis auf die Zielklausel in den `usable`-Teil stellen.

(d) Wenn Sie alles richtig gemacht haben, wird Ihnen `otter` in der vorigen Teilaufgabe die Auskunft gegeben haben, dass die Formel unerfüllbar ist, d.h. es gibt eine Lösung. Interessant ist freilich auch, diese Lösung zu kennen. Erweitern wir dazu das *erreichbar*-Prädikat um eine fünfte Stelle. Diese soll eine Liste enthalten, die die Schritte angibt, mit denen die jeweilige Situation erreicht wurde.

Um Listen in `otter` darzustellen, verwenden wir Funktionssymbole. Die Konstante e drückt die leere Liste aus, mit $c(L, x)$ drücken wir die Liste L gefolgt von dem Element x aus. Beispiel: $c(c(e, a), b)$ ist eine Liste, die aus a gefolgt von b besteht.

Unsere Tatsachenklausel erweitern wir wie folgt:

$$erreichbar(l, l, l, l, e)$$

Erweitern Sie nun die Prozedurklauseln so, dass zu jeder erreichbaren Situation die Schritte berechnet werden, mit denen sie erreicht wird. Als Listenelemente verwenden Sie z.B. Konstanten wie *nurMann* (der Mann überquert den Fluss allein), *mitWolf* (der Mann überquert den Fluss zusammen mit dem Wolf) etc.

Führen Sie das entsprechend erweiterte `otter`-Programm aus. Erweitern Sie die Zielklausel dazu wie folgt:

$$\neg erreichbar(r, r, r, r, L) \vee \$ans(L)$$

(Dadurch wird errreicht, dass `otter` den Wert von L als Ergebnis ausgibt.) Führen Sie den Erfüllbarkeitstest erneut mit und ohne Stützmengenrestriktion durch.

Aufgabe 1 **(Lösungsvorschlag)** *Modellieren mit Prolog*

(a) Zunächst definieren wir, was es bedeutet, dass zwei Dinge an gegenüberliegenden Ufern stehen:

$$\{gegenueber(l, r)\}$$
$$\{gegenueber(r, l)\}$$

Das Wolf/Ziege-Problem ist entschärft, wenn Wolf und Ziege an unterschiedlichen Ufern stehen oder der Mann beim Wolf ist:

$$\{\neg gegenueber(W, Z), wolf_ok(M, W, Z, K)\}$$
$$\{wolf_ok(M, M, Z, K)\}$$

Analog das Ziege/Kohlkopf-Problem:

$$\{\neg gegenueber(Z, K), ziege_ok(M, W, Z, K)\}$$
$$\{ziege_ok(M, W, M, K)\}$$

(b) Befindet sich der Mann an einem Ufer, so kann er alleine ans gegenüberliegende Ufer fahren, sofern die entstehende Situation keine Gefahren birgt. Alle anderen Objekte bleiben am selben Ufer.

$$\{\neg erreichbar(M, W, Z, K), \neg gegenueber(M, Mn), \neg ok(Mn, W, Z, K),$$
$$erreichbar(Mn, W, Z, K)\}$$

Befinden sich Mann und Wolf (bzw. Ziege oder Kohlkopf) am selben Ufer, so können beide übersetzen, wenn die entstehende Situation keine Gefahren birgt.

$$\{\neg erreichbar(M, M, Z, K), \neg gegenueber(M, Mn), \neg ok(Mn, Mn, Z, K),$$
$$erreichbar(Mn, Mn, Z, K)\}$$
$$\{\neg erreichbar(M, W, M, K), \neg gegenueber(M, Mn), \neg ok(Mn, W, Mn, K),$$
$$erreichbar(Mn, W, Mn, K)\}$$
$$\{\neg erreichbar(M, W, Z, M), \neg gegenueber(M, Mn), \neg ok(Mn, W, Z, Mn),$$
$$erreichbar(Mn, W, Z, Mn)\}$$

(c) Aller Wahrscheinlichkeit wird man herausbekommen, dass es `otter` ohne Stützmengen-Restriktion nicht gelingt, den Beweis zu finden. Benutzt man die Restriktion, so wird der Beweis in weniger als einer Sekunde gefunden.

(d) Als Beispiel sei die modifizierte Regel gegeben für den Fall, dass der Mann den Fluss allein überquert; die übrigen Modifikationen sind analog.

$$\{\neg erreichbar(M, M, Z, K, L), \neg gegenueber(M, Mn), \neg ok(Mn, Mn, Z, K), \\ erreichbar(Mn, Mn, Z, K, c(L, nurMann))\}$$

Auch in der modifizierten Variante wird `otter` nur mit der Restriktion erfolgreich sein. Die Ausgabe ist die Sequenz folgender Züge:

mitZiege, *nurMann*, *mitWolf*, *mitZiege*, *mitKohlkopf*, *nurMann*, *mitZiege*