

Prof. Dr. Jürgen Giesl
Peter Schneider-Kamp
René Thiemann

Übungen *Logikprogrammierung* – Blatt 14

Abgabe bis Mittwoch, 19. Juli 2006, im Übungskasten.

Infos zur Scheinklausur am 14.7.:

- In der Klausur sind weder Unterlagen noch andere Hilfsmittel zugelassen.
- Die Klausur findet in der Zeit von 10.00 Uhr bis 11.30 Uhr statt.
- Hörsaalteilung: Studierende mit Matrikel-Nr. ≤ 243000 : AH II
> 243000: AH III

Aufgabe 1 (1,5 + 3 + 4,5 Punkte)

Ziel der Aufgabe ist es, ein Prädikat `rate/0` zu definieren, welches den Benutzer ein Tier ausdenken lässt, welches das System dann durch ja/nein-Fragen erraten soll. Falls dies nicht gelingt, soll die interne Wissensbasis erweitert werden, so dass das Tier beim nächsten Versuch erraten werden kann. Dies geschieht, indem der Benutzer nach einem Unterscheidungsmerkmal gefragt wird. Ein Protokoll könnte wie folgt aussehen (*Systemfont*, *Benutzerfont*):

?- *rate*.

Denken Sie sich ein Tier aus...

Lebt es im Meer? *noe*.

'noe' ist keine gueltige Antwort! Lautet die Antwort 'ja' oder 'nein'? *nein*.

Ist Ihr Tier ein Schlange? *nein*.

Leider kenne ich Ihr Tier nicht. Welches Tier haben Sie sich ausgedacht? '*Hund*'.

Damit ich nicht dumm bleibe, geben Sie bitte eine Frage ein,

die fuer Ihr Tier zutrifft, aber nicht fuer ein/eine/einen Schlange: '*Hat es Pfoten?*'.

Danke sehr, vielleicht schaffe ich es ja beim naechsten Mal.

?- *rate*.

Denken Sie sich ein Tier aus...

Lebt es im Meer? *nein*.

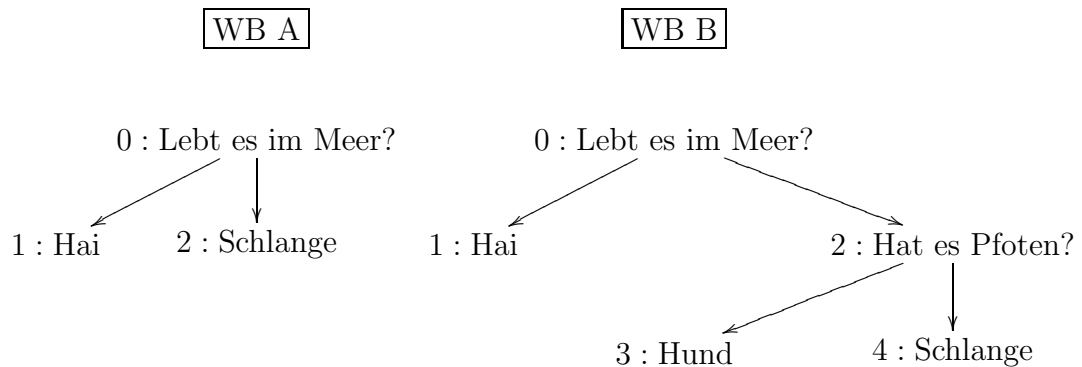
Hat es Pfoten? *ja*.

Ist Ihr Tier ein Hund? *ja*.

Ich habe es gewusst!

- Implementieren Sie ein Prädikat `getNextNr/1`, welches nach und nach die Zahlen 1,2,...liefert. Die Anfrage `?- getNextNr(X), getNextNr(Y)` sollte also mit `X/1, Y/2` beantwortet werden. Stellt man die Anfrage nochmal, so bekommt man die Antwort `X/3, Y/4`.

- Die Wissensbasis des Systems lässt sich als ein Binärbaum darstellen, der Tiere in den Blättern, und Fragen in den inneren Knoten enthält.



Nutzen Sie zur Darstellung der Wissensbasis ein dynamisches Prädikat `wb/2`. Diese repräsentiert einen Ratebaum wie in der Abbildung. Das erste Argument von `wb` ist eine Knotennummer, das zweite beinhaltet entweder das geratene Tier oder die Unterscheidungsfrage sowie die Nummern der beiden Unterbäume.

Erstellen Sie ein Prädikat `updateWB(N, New, Question)`, welches das bisherige Blatt mit Nummer `N` und einem Tier `Old` durch einen Frageknoten mit Frage `Question` ersetzt. Von dem neuen Frageknoten soll zudem auf zwei neue Blätter (mit neuen Nummern) mit den Tieren `New` und `Old` verwiesen werden.

Falls z.B. im Prädikat `wb` der Baum WB A kodiert ist, dann sollte nach der Anfrage `?- updateWB(2, 'Hund', 'Hat es Pfoten?')` der Baum WB B im Prädikat `wb` kodiert sein.

Initialisieren Sie Ihre Wissensbasis gemäß WB A.

- Implementieren Sie das Prädikat `rate/0`.

Aufgabe 2 (3 Punkte)

Gegeben sei die folgende Grammatik G über den Terminalsymbolen $\{x,y,z,0,1,(,),\times,+\}$ mit dem Startsymbol `Poly`. Hierbei bezeichnet ε das leere Wort.

Poly \rightarrow Variable	DigitSequence $\rightarrow \varepsilon$
Poly \rightarrow Number	DigitSequence $\rightarrow 0$ DigitSequence
Poly \rightarrow Poly \times Poly	DigitSequence $\rightarrow 1$ DigitSequence
Poly \rightarrow Poly $+$ Poly	Variable $\rightarrow x$
Poly \rightarrow (Poly)	Variable $\rightarrow y$
Number $\rightarrow 0$	Variable $\rightarrow z$
Number $\rightarrow 1$ DigitSequence	

Schreiben Sie ein Prädikat `poly/1`, so dass bei Darstellung von Wörtern als Listen `poly(w)` genau dann wahr ist, wenn `w` von G erkannt wird. (Das Wort "10+x" entspricht der Liste `[1,0,'+',x]`.) Ihr Prolog Programm darf weder das Symbol `-->`, noch Prädikate zur Listenkonkatenation enthalten. Sie sollten soweit wie möglich Differenzlisten benutzen.