

## 2. Übungszettel Künstliche Intelligenz SS15

Prof. Raúl Rojas, Dr. Christoph Benzmüller, Fritz Ulbrich  
Institut für Informatik, Freie Universität Berlin  
Abgabe bis Freitag, 08.05.15, 14.00 Uhr

### 1. Aufgabe (6 Punkte): TPTP

Machen Sie sich mit der TPTP-Sprache für FOF (first-order form, Logik erster Stufe) vertraut (<http://www.cs.miami.edu/~tptp/TPTP/QuickGuide/>). Beschreiben Sie die folgenden (lösbaren) Probleme in TPTP-Sprache (in Logik erster Stufe, fof) und lassen es von einem Theorembeweiser für FOF (First Order Format) unter <http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP> lösen.

- (1 Punkt) **Axioms:** If you work hard then you get lucky. Either you get lucky or you work hard, or both. If you get lucky then, either you are not a rogue or you work hard (but not both). You are a rogue. **Conjecture:** You work hard.
- (2 Punkte) **Axioms:** Garfield is a cat. Odie is a dog. Cats and dogs are animals. Jon is a human. Every animal has a human owner. Jon is the owner of Garfield and Odie. Garfield and Odie are the only animals that Jon owns. If a cat is chased by a dog, then the owner of the cat hates the owner of the dog. Odie has chased Garfield. **Conjecture:** Jon hates himself.
- (3 Punkte) **Axioms:** Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants. **Conjecture:** There is an animal that likes to eat a grain eating animal.

Hinweis: Sie erhalten eventuell sinnvollere bzw. lesbarere Hinweise, warum ein Beweis fehlgeschlagen ist, wenn Sie das Problem von einem "model finder" (z.B. Metris, E, DarwinFM) lösen lassen.

### 2. Aufgabe (4 Punkte): Listen in Prolog

Geben Sie die Definitionen für folgende Regeln an:

- (1 Punkt) **myLast(List, LastItem):** LastItem ist das letzte Element in List.
- (1 Punkt) **myMax(ListOfNumbers, MaxItem):** MaxItem ist die größte Zahl in ListOfNumbers.
- (1 Punkt) **mySum(ListOfNumbers, SumList):** SumList entspricht der Summe der Zahlen in ListOfNumbers.
- (1 Punkt) **myOrdered(ListOfNumbers)** ist erfüllt, wenn die Liste sortiert ist(aufsteigend oder absteigend).

Hilfreiche Prolog-Ausdrücke:

**[]:** leere Liste.

**[\_]:** Liste mit einem (beliebigen) Element.

**[H|T]:** Liste mit erstem Element H und Restliste T.

**[H1,H2|T]:** Liste mit ersten Elementen H1 und H2 und Restliste T.

**A is B:** Ist wahr, wenn der ausgewertete Ausdruck B gleich A ist, z.B. 6 is 2 + 5 - 1.

**A >= B:** Ist wahr, wenn Zahl A größer oder gleich B