

Übungsblatt 11

Abgabe der Lösungen: Tutorium in der Woche 15.07-19.07

Präsenzaufgabe: Fitch mit Quantoren (0 Punkte)

Beweisen Sie die folgenden Formeln mit Hilfe des Fitch-Kalküls mit Quantoren:

1. $((\forall X. p(X) \rightarrow p(f(X))) \wedge p(a)) \rightarrow p(f(f(a)))$;
2. $(\forall X. \exists Y p(X, Y)) \vee (\exists X. \forall Y \neg p(X, Y))$.

Aufgabe 1 Fitch mit Quantoren (8 Punkte)

Beweisen Sie die folgenden Formeln mit Hilfe des Fitch-Kalküls mit Quantoren:

1. $(\exists X. (p(X) \rightarrow \neg p(f(X)))) \rightarrow \exists X. \neg p(X)$;
2. $(\forall X. p(X) \vee \forall X. q(X)) \rightarrow \forall X. (p(X) \vee q(X))$.

Aufgabe 2 Suchen in Wissensbasen (12 Punkte)

Wir betrachten hier eine Prolog-Implementierung einer Wissensbasis über Personen, die als binärer Suchbaum organisiert ist. Die Wissensbasis besteht aus Fakten (d.h. Regeln ohne Voraussetzungen) zweierlei Typs:

- Fakten der Form `query(frage, kategorie, t_kategorie, f_kategorie)` repräsentieren *innere Knoten* des Baums, wobei die Argumente wie folgt zu verstehen sind:
 - `frage` ist eine natürlichsprachlich formulierte Ja/Nein-Frage, die die angegebene `kategorie` weiter verfeinert.
 - `kategorie` ist eine Bezeichnung für eine Gruppe von Personen. Der Name `root` ist dabei für die allgemeinste Kategorie, also die Gruppe *aller* Personen, reserviert und bildet die Wurzel des Baums.
 - `t_kategorie` und `f_kategorie` bezeichnen die Unterkategorien von `kategorie`, auf die bei positiver bzw. negativer Antwort auf die `frage` zu schließen ist.
- Fakten der Form `done(kategorie, name)` repräsentieren *Randknoten* des Baums und verknüpfen Kategorien mit Personennamen, nach dem Motto „ich kenne nur einen einzigen dicken Sänger aus dem letzten Jahrhundert, also handelt es sich bei der gesuchten Person um Nuccio Ravapotti“

Beispiel: Die folgende Wissensbasis umfasst drei Personen John (Mann), Heidi (Frau, blaue Augen) und Marta (Frau, grüne Augen).

```

1 query('Is it a man?\n', root, m, f).
2 query('Does she have blue eyes?\n', f, fb, fg).
3
4 done(m, john).
5 done(fb, heidi).
6 done(fg, marta).
```

1. Entwerfen Sie ein Wissensbasis, die mindestens acht Personen oder Wesen aus Ihrer Interessensphäre umfasst (z.B. Freunde, Schauspieler, Sportler, Star-Trek-Figuren, etc.).
2. Erweitern Sie Ihr Programm um ein Prädikat `whois/1`, das eine interaktive Suche in der Wissensdatenbank durchführt, indem es dem Nutzer Fragen stellt, bis die gesuchte Person identifiziert ist. Das heißt, die Anfrage `whois(X)` muss mit `X=name` beantwortet werden. Verwenden Sie dabei die Prolog-eingebauten Prädikate `read/1` und `write/1` zum Lesen bzw. Schreiben auf Konsole.

Beispiel des beabsichtigten Programmablaufs:

```

1 ?- [id].
2 % id compiled 0.00 sec, 11,320 bytes
3 true.
4
5 ?- whois(X).
6 Think of a character from Alice in Wonderland..
7 Is it an animal like one found in nature?
8 |: yes.
9 Not in range. Please, try again.
10 |: y.
11 Does it have a long tail?
12 |: y.
13 Does it tend to raise philosophical points?
14 |: y.
15 Is it known for its mysterious behaviour?
16 |: y.
17 X = cheshire_cat .
```

Bonusaufgabe

(2 Punkte)

Was ist die kleinstmögliche Anzahl von Fragen, die der Nutzer beantworten muss, um jede Person in der Wissensbasis zu identifizieren, mit der Maßgabe, dass die Wissensdatenbank insgesamt N Personen abspeichert? (Genauer gesagt suchen wir die kleinste Funktion $g(N)$, so dass eine Wissensbasis mit N Personen so aufgebaut werden kann, dass jede Person mit höchstens $g(N)$ Fragen gefunden wird.) Begründen Sie ihre Antwort.