Artificial Intelligence
Exercise 7

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

9. Februar 2017

Die Lösungen bitte als python-Datei (siehe Vorlage in der Email/Website) mit dem Namen `e07/e07_sol.py` (in Verzeichnis `e07`) in Euer git account einloggen. In der python-Datei wird das Format, in dem Antworten gegeben werden sollen, genauer erklärt. Bei Unklarheiten bitte bei dem Tutor melden.

Diese letzte Übung zählt zu den Programmieraufgaben. Dies ist keine Bonusaufgabe. Präsenz- und Votieraufgaben gibt es nicht mehr.

Abgabe bis Montag, 20. Februar.

1 Erfüllbarkeit und allgemeine Gültigkeit (Aussagenlogik) (30%)

Entscheiden Sie, ob die folgenden Sätze erfüllbar (*satisfiable*), allgemein gültig (*valid*) oder keins von beidem (*none*) sind.

(a) $Smoke \Rightarrow Smoke$

(b) $Smoke \Rightarrow Fire$

(c) $(Smoke \Rightarrow Fire) \Rightarrow (\neg Smoke \Rightarrow \neg Fire)$

(d) $Smoke \vee Fire \vee \neg Fire$

(e) $((Smoke \wedge Heat) \Rightarrow Fire) \Leftrightarrow ((Smoke \Rightarrow Fire) \vee (Heat \Rightarrow Fire))$

(f) $(Smoke \Rightarrow Fire) \Rightarrow ((Smoke \wedge Heat) \Rightarrow Fire)$

(g) $Big \vee Dumb \vee (Big \Rightarrow Dumb)$

(h) $(Big \wedge Dumb) \vee \neg Dumb$

2 Modelle enumerieren (Aussagenlogik) (30%)

Betrachten Sie die Aussagenlogik mit Symbolen A, B, C und D. Insgesamt existieren also 16 Modelle. In wievielen Modellen sind die folgenden Sätze erfüllt?

(a) $(A \wedge B) \vee (B \wedge C)$

(b) $A \vee B$

(c) $A \Leftrightarrow (B \Leftrightarrow C)$

3 Unifikation (Prädikatenlogik) (40%)

Geben Sie für jedes Paar von atomaren Sätzen den allgemeinsten Unifikator an, sofern er existiert. Standardisieren Sie nicht weiter. Geben Sie None zurück, wenn kein Unifikator existiert. Ansonsten ein Dictioary, dass als Key die Variable und als Value die Konstante enthält.

Für $P(A), P(x)$: `sol3z = {'x': 'A'}`

(a) $P(A, B, B), P(x, y, z)$.

(b) $Q(y, G(A, B)), Q(G(x, x), y)$.

(c) $Older(Father(y), y), Older(Father(x), John)$.

(d) $Knows(Father(y), y), Knows(x, x)$.

4 Privat-Spaß-Aufgabe: as Constraint Satisfaction Problem

Consider the Generalized Modus Ponens (slide 09:15) for inference (forward and backward chaining) in first order logic. Applying this inference rule requires to find a substitution θ such that $p_i'\theta = p_i\theta$ for all i.

Show constructively that the problem of finding a substitution θ (also called **matching problem**) is equivalent to a Constraint Satisfaction Problem. "Constructively" means, explicitly construct/define a CSP that is equivalent to the matching problem.

Note: The PDDL language to describe agent planning problems (slide 08:24) is similar to a knowledge in Horn form. Checking whether the action preconditions hold in a given situation is exactly the matching problem; applying the Generalized Modus Ponens corresponds to the application of the action rule on the current situation.

5 Privat-Spaß-Aufgabe

In the lecture we discussed the case
"A first cousin is a child of a parent's sibling"

$$\forall x, y \; FirstCousin(x, y) \iff \exists p, z \; Parent(p, x) \wedge Sibling(z, p) \wedge Parent(z, y)$$

A question is whether this is equivalent to

$$\forall x, y, p, z \; FirstCousin(x, y) \iff Parent(p, x) \wedge Sibling(z, p) \wedge Parent(z, y)$$

Let's simplify: Show that the following two

$$\forall x \; A(x) \iff \exists y \; B(y, x) \tag{1}$$

$$\forall x, y \; A(x) \iff B(y, x) \tag{2}$$

are different. For this, bring both sentences in CNF as described on slides 09:21 and 09:22 of lecture 09-FOLinference.