

## Blatt 5 (8 Punkte)

Abgabe durch Hochladen auf der eCampus-Seite bis Sonntag, 15.05.2016, 23:59 Uhr, in Gruppen von 2-3 Personen.

*Beachten Sie, dass von Aufgaben 5.1 und 5.2 nur eine auszusuchen ist, die abgegeben werden kann/muss. Es wird dementsprechend nur eine Aufgabe bepunktet korrigiert. Bei Abgabe beider Aufgaben bitte kenntlich machen, welche bewertet werden soll.*

### Aufgabe 5.1: Programmieraufgabe: Resolution

(entweder hier 3)

**Vorbereitung:** Laden Sie bitte das ZIP-Archiv logic.zip herunter von unserer eCampus-Seite unter Kursunterlagen >> Python und AIMA Python >> AIMA-Py Logic. Das ZIP-Archiv enthält das Skript AIMA.py und den Ordner logic mit den Skripten logic.py und Findresolvents.py, sowie eine vorgegebene Klauselmenge BeispielVorlesung.cls. Ersetzen Sie Ihr bisher verwendetes Skript AIMA.py im Ordner aima durch das neu heruntergeladene aktualisierte AIMA.py und fügen Sie den Ordner logic auf der gleichen Ebene wie den Ordner aima ein.

**Aufgabe:** Ergänzen Sie Findresolvents.py so, dass hier alle Resolventen der zwei Input-Hornklauseln clauseA und clauseB gefunden werden und in newclauses als Liste gespeichert werden. Testen Sie das von Ihnen veränderte Skript, z.B. indem Sie BeispielVorlesung.cls als Wissensbasis verwenden, logic.py als Algorithmus laden und EvilJohn als Ziel eingeben.

### Aufgabe 5.2: PL1-Resolution

(oder hier 1 + 2 = 3)

a) Betrachten Sie folgende Sätze und ihre Übersetzungen in Formeln der Prädikatenlogik 1. Stufe. Wandeln Sie diese Formeln in KNF-Klauselform um (Vorlesung 7, Folie 32f).

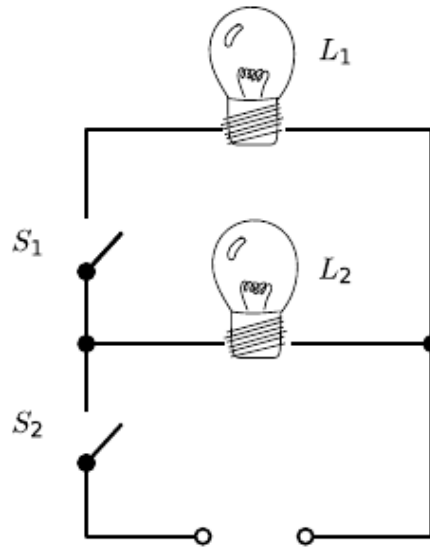
- 1) Alle Katzen sind niedlich und alle Hunde sind treu.  
 $\forall x_1 \text{ Katze}(x_1) \Rightarrow \text{Niedlich}(x_1) \wedge \forall x_2 \text{ Hund}(x_2) \Rightarrow \text{Treu}(x_2)$
- 2) Bello, Kiki, Mausi und Struppi sind meine Haustiere.  
 $\text{Haustier}(b) \wedge \text{Haustier}(k) \wedge \text{Haustier}(m) \wedge \text{Haustier}(s)$
- 3) Alle meine Haustiere sind Hunde oder Katzen.  
 $\forall x_3 \text{ Haustier}(x_3) \Rightarrow (\text{Hund}(x_3) \vee \text{Katze}(x_3))$
- 4) Wenn Mausi niedlich ist, dann ist Kiki nicht treu.  
 $\text{Niedlich}(m) \Rightarrow \neg \text{Treu}(k)$
- 5) Wenn Kiki niedlich ist, dann ist Struppi nicht niedlich.  
 $\text{Niedlich}(k) \Rightarrow \neg \text{Niedlich}(s)$
- 6) Mausi ist eine Katze.  
 $\text{Katze}(m)$

b) Zeigen Sie durch Widerlegung des Gegenteils mittels Resolution:

- 1) “Kiki ist eine Katze”,
- 2) “Struppi ist ein Hund”, wobei Sie die Klausel “Kiki ist eine Katze” nutzen dürfen.

### Aufgabe 5.3: Situationskalkül

(2 + 0.5 = 2.5)



Formalisieren Sie die oben dargestellte Parallelschaltung mit Hilfe des Situationskalküls. Hierbei sind  $S_1$  und  $S_2$  Schalter, die nur Strom fließen lassen, wenn diese geschlossen sind. Die Lampen  $L_1$  und  $L_2$  leuchten natürlich nur, wenn Strom auf der entsprechenden Leitung fließt. Es gibt folgende Fluente:

- $\text{leuchtet}(L, s)$  gilt, wenn die Lampe  $L$  in Situation  $s$  leuchtet, also mit Strom versorgt wird.
- $\text{geschlossen}(X, s)$  gilt, wenn der Schalter  $X \in \{S_1, S_2\}$  in Situation  $s$  geschlossen ist.

Außerdem gibt es die Aktionen  $\text{schalte}(X)$  zum Öffnen und Schließen eines Schalters. D.h. ein Schalter, der geschlossen ist in Zustand  $s_t$ , ist nach Ausführen der Aktion  $\text{schalte}(X)$  im Nachfolgezustand  $s_{t+1} = \text{do}(\text{schalte}(X), s_t)$  wieder geöffnet.

- a) Geben Sie für die folgenden Fluente jeweils eine vollständige Beschreibung des Nachfolgezustandes durch Nachfolgezustandsaxiome (Vorlesung 8) an.

- 1)  $\text{geschlossen}(S_1, s)$
- 2)  $\text{leuchtet}(L_1, s)$

Für den Fluente  $\text{leuchtet}(L_2, s)$  wäre dies z.B.:

$$\forall a, s : \text{leuchtet}(L_2, \text{do}(a, s)) \Leftrightarrow [(a = \text{schalte}(S_2) \wedge \neg \text{geschlossen}(S_2, s)) \vee (a = \text{schalte}(S_1) \wedge \text{geschlossen}(S_2, s))]$$

- b) Beschreiben Sie den Anfangszustand  $s_0$  als Term im Situationskalkül, in dem alle Schalter offen sind.

#### Aufgabe 5.4: STRIPS

(2 + 0.5 = 2.5)

Gegeben sei folgendes Planungsproblem im STRIPS Formalismus:

- Fluenzen: fit, vorbereitet
- Operatoren:
  - Op ( Action: Lernen, Precond:  $\neg$ vorbereitet, Effects: vorbereitet  $\wedge$   $\neg$  fit)
  - Op ( Action: Schlafen, Precond:  $\emptyset$ , Effects: fit)
- Start:  $\neg$ vorbereitet
- Ziel: fit, vorbereitet

Konstruieren Sie den partiell geordneten Plan mittels des POP-Algorithmus mit Tiefensuche:

- a) Zeichnen Sie den fertigen partiell geordneten Plan als Diagramm inklusive aller kausaler Kanten als Doppelpfeile " $\Rightarrow$ " und Ordnungsrelationen als einfache Pfeile " $\rightarrow$ ". Ordnungsrelationen müssen nur dann explizit zwischen zwei Knoten eingezeichnet werden, wenn sonst noch kein (gerichteter) kausaler Pfad zwischen diesen Knoten existiert. Es ist ein Promotions/Demotions-Schritt durchzuführen. Heben Sie diesen besonders hervor und geben sie an, ob es sich um Promotion oder Demotion handelt.
- b) Leiten Sie aus ihrem partiell geordneten Plan im Anschluß alle möglichen vollständig geordneten Pläne ab.