

## Beschreibungslogik

### Übungsblatt 3

Abgabe im PDF-Format bis 21. 5. 2018, 23:59 Uhr in Stud.IP, Ordner „Abgabe Übungsblatt 3“  
Bitte nur eine PDF-Datei pro Gruppe, Lizenz „Selbst verfasstes, nicht publiziertes Werk“.

---

1. (20 %) Verwende den Tableau-Algorithmus für  $\mathcal{ALC}$  aus der Vorlesung, um die Erfüllbarkeit der folgenden Konzepte zu entscheiden.

a)  $C_0 = \exists r.A \sqcap \exists r.B \sqcap \forall r.\exists r.A \sqcap \forall r.\exists r.\neg A \sqcap \forall r.\forall r.\neg A$

b)  $C_0 = \neg(\forall r.(A \sqcup B) \sqcap \forall r.(A \sqcup \neg B)) \sqcap \neg\exists r.(\neg A \sqcap \neg B)$

Gib an, welche Regeln in welcher Reihenfolge worauf angewendet werden. Gib im Falle von Erfüllbarkeit außerdem ein Modell gemäß dem Beweis von Theorem 4.8 an.

2. (20 %) Im Computerspiel *Multimonster Massacre* bist Du in einem Raum mit  $n \geq 0$  Monstern gefangen. Jedes Monster hat bis zu 100 Köpfe. Du hast eine unbegrenzte Anzahl von Schlägen zur Verfügung. Mit jedem Schlag erlegst Du genau ein Monster, welches dann verschwindet und sofort durch eine beliebige Anzahl neuer Monster ersetzt wird, von denen aber jedes strikt weniger Köpfe hat als das erlegte. Ein kopfloses Monster verwandelt sich sofort in einen Keks.

Zeige mittels einer Multimengenordnung, dass nach endlich vielen Schlägen nur noch Kekse im Raum sind.

3. (20 %) Verwende den Tableau-Algorithmus für  $\mathcal{ALC}$  mit TBoxen aus der Vorlesung um zu entscheiden, ob

a)  $C_1 = \exists r.(\exists r.A \sqcup \exists r.B)$  erfüllbar bzgl.  $\mathcal{T}_1 = \{\top \sqsubseteq \forall r.(\neg A \sqcap \neg B)\}$  ist;

b)  $C_2 = \neg A \sqcap B \sqcap \forall r.(B' \sqcap \forall r.\neg A)$  erfüllbar bzgl.  $\mathcal{T}_2 = \{\top \sqsubseteq \exists r.B \sqcap \exists s.B\}$  ist;

c) die Subsumtion  $\mathcal{T}_3 \models \text{Student} \sqsubseteq \text{Happy}$  gilt,  
wobei  $\mathcal{T}_3 = \{\text{Student} \sqsubseteq \exists \text{solves.Exercise}, \exists \text{solves.}\top \sqsubseteq \text{Happy}\}$ .

Im Falle von Erfüllbarkeit bzw. Nicht-Subsumtion gib das Modell an, das sich aus der Konstruktion im Beweis von Theorem 4.13 ergibt.

4. (20 %) Benutze Protégé 5 und einen Reasoner (z. B. **Hermit**, **FaCT++** oder **Pellet**), um Deine Ergebnisse aus Aufgabe 3 zu überprüfen. Schreibe dazu jeweils eine OWL-Ontologie mit den Konzeptinklusionen aus  $\mathcal{T}_i$  und verfare wie folgt.

a) Füge zu  $\mathcal{T}_1$  die Definition  $C_1 \equiv \exists r.(\exists r.A \sqcup \exists r.B)$  hinzu. Klassifiziere die Ontologie. In der *Class hierarchy* wechsele zu *Inferred* und schaue nach, ob  $C_1$  rot eingefärbt ist und unter **owl:Nothing** erscheint (genau dann ist  $C_1$  unerfüllbar bzgl.  $\mathcal{T}_1$ ).

b) Verfahre genauso für  $C_2$  und  $\mathcal{T}_2$ .

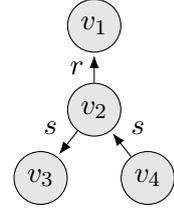
c) Klassifiziere  $\mathcal{T}_3$  und schaue in der *Class hierarchy (Inferred)* nach, ob **Student** unter **Happy** erscheint. Hinweis: zum Modellieren der Konzeptinklusion  $\exists \text{solves.}\top \sqsubseteq \text{Happy}$  wähle die *Object Property solves* und füge unter *Domains (intersection)* die Klasse **Happy** hinzu.

Lege Deiner Abgabe die drei Ontologien und je einen Screenshot der *Class hierarchy (Inferred)* nach dem Klassifizieren bei (bitte alles in eine ZIP-Datei packen).

Benutze bei Bedarf die in Aufgabe 5 b) auf Blatt 1 genannten Hilfsmittel.

Bitte wenden.

5. (20%) Betrachte die folgende Erweiterung des Tableau-Algorithmus mit TBoxen aus der Vorlesung auf die Beschreibungslogik  $\mathcal{ALCI}$ :



1. Erweiterung von I-Bäumen:

- Kanten dürfen vorwärts und rückwärts zeigen; im nebenstehenden Beispiel ist  $E = \{(v_2, r, v_1), (v_2, s, v_3), (v_4, s, v_2)\}$ .

2. zusätzliche  $\exists^-$ -Regel:

- Wähle  $v \in V$  und  $\exists r^-.C \in \mathcal{L}(v)$ , so dass es kein  $v' \in V$  gibt mit  $(v', r, v) \in E$  und  $C \in \mathcal{L}(v')$ .
- Erweitere  $V$  um neuen Knoten  $v'$  und  $E$  um  $(v', r, v)$ , und setze  $\mathcal{L}(v') = \{C\}$ .

3. zusätzliche  $\forall^-$ -Regel:

- Wähle  $v, v' \in V$  und  $\forall r^-.C \in \mathcal{L}(v)$ , so dass  $(v', r, v) \in E$  und  $C \notin \mathcal{L}(v')$ .
- Erweitere  $\mathcal{L}(v')$  um  $C$ .

Zeige, dass der skizzierte Algorithmus auf der folgenden Eingabe ein falsches Ergebnis liefert:  $C_0 = A$  und  $\mathcal{T} = \{\top \sqsubseteq \exists r.A, \top \sqsubseteq \forall r^-. \forall r^-. \neg A\}$

6. **Zusatzaufgabe** (20%) Mit  $\mathcal{ALC}_{\text{trans}}$  bezeichnen wir die Erweiterung von  $\mathcal{ALC}$  um transitive Rollen, d. h. die TBox darf nun zusätzlich zu Konzeptinklusionen auch Zusicherungen der Form  $\text{trans}(r)$  enthalten, wobei  $r$  ein Rollenname ist. Eine Interpretation  $\mathcal{I}$  erfüllt  $\text{trans}(r)$ , wenn  $r^{\mathcal{I}}$  eine transitive Relation ist.

Sei  $\mathcal{T}$  eine  $\mathcal{ALC}_{\text{trans}}$ -TBox der Form  $\{\top \sqsubseteq C_{\mathcal{T}}\}$  mit  $C_{\mathcal{T}}$  in NNF. Wir definieren eine  $\mathcal{ALC}$  TBox  $\mathcal{T}^*$  wie folgt:

- $\mathcal{T}^*$  enthält  $\top \sqsubseteq C_{\mathcal{T}}$ .
- Für jedes  $\forall r.C \in \text{sub}(C_{\mathcal{T}})$  mit  $\text{trans}(r) \in \mathcal{T}$  enthält  $\mathcal{T}^*$  die Konzeptinklusion  $\forall r.C \sqsubseteq \forall r. \forall r.C$ .

- a) Beweise, dass für alle Konzeptnamen  $A$  gilt:

$A$  ist erfüllbar bzgl.  $\mathcal{T}$  gdw.  $A$  ist erfüllbar bzgl.  $\mathcal{T}^*$

- b) Wie kann man diese Reduktion nutzen, um mit Hilfe des Tableau-Algorithmus für  $\mathcal{ALC}$  Erfüllbarkeit in  $\mathcal{ALC}_{\text{trans}}$  (mit TBoxen) zu entscheiden? Erkläre kurz.