Prof. Dr. Thomas Schneider SoSe 2018

Beschreibungslogik

Fragebogen 11 vom 16. 5.

---

1. Eliminierung schlechter Typen

Warum sind im Beispiel zu Folie 11 (T 5.1) die Typen t_0, t_1, t_2 schlecht in Γ_1?

---

2. Korrektheit Typelimination

Wo finden sich im Beweis von Lemma 5.5 die Eigenschaften eines Typs (siehe unten) und die Abwesenheit von schlechten Typen wieder?

1. ______
2. ______
3. ______
4. ______

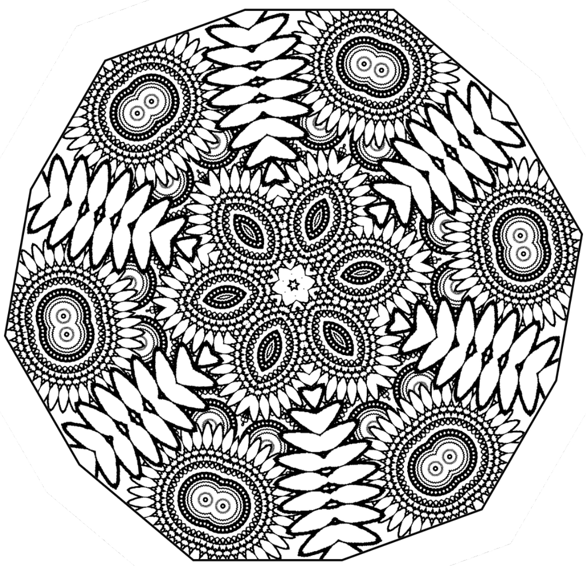
- „Γ_i enthält keine schlechten Typen mehr“:

______

Zur Erinnerung hier nochmal die 4 Eigenschaften von Typen:

Ein *Typ für* C_0 *und* $\mathcal{T}$ ist eine Teilmenge $t \subseteq \mathsf{sub}(C_0, \mathcal{T})$, so dass

1. $A \in t$ gdw. $\neg A \notin t$ für alle $\neg A \in \mathsf{sub}(C_0, \mathcal{T})$
2. $C \sqcap D \in t$ gdw. $C \in t$ und $D \in t$ für alle $C \sqcap D \in \mathsf{sub}(C_0, \mathcal{T})$
3. $C \sqcup D \in t$ gdw. $C \in t$ oder $D \in t$ für alle $C \sqcup D \in \mathsf{sub}(C_0, \mathcal{T})$
4. $C_{\mathcal{T}} \in t$



Bitte wenden.

Syntaktische Typen

Wir nehmen an, dass

- das Eingabe-Konzept C_0 in NNF ist und
- die Eingabe-TBox die Form $\{\top \sqsubseteq C_\mathcal{T}\}$ hat mit $C_\mathcal{T}$ in NNF.

Definition 5.2 (Typ)

Ein **Typ für C_0 und $\mathcal{T}$** ist eine Teilmenge $t \subseteq \mathrm{sub}(C_0, \mathcal{T})$, so dass

1. $A \in t$ gdw. $\neg A \notin t$ für alle $\neg A \in \mathrm{sub}(C_0, \mathcal{T})$
2. $C \sqcap D \in t$ gdw. $C \in t$ und $D \in t$ f. alle $C \sqcap D \in \mathrm{sub}(C_0, \mathcal{T})$
3. $C \sqcup D \in t$ gdw. $C \in t$ oder $D \in t$ f. alle $C \sqcup D \in \mathrm{sub}(C_0, \mathcal{T})$
4. $C_\mathcal{T} \in t$

T 5.1

Schlechte Typen

Präzisierung von „Typen, die in keinem Modell vorkommen können“:

Definition 5.3 (schlechter Typ)

Sei Γ Typenmenge und $t \in \Gamma$.

Dann ist t **schlecht in Γ**, wenn für ein $\exists r.C \in t$ gilt:

Es gibt kein $t' \in \Gamma$ mit $\{C\} \cup \{D \mid \forall r.D \in t\} \subseteq t'$.

T 5.1 Forts.

Intuitiv:
Typ ist schlecht, wenn er eine existentielle Restriktion enthält, für die es keinen „Zeugen“ gibt.

Typelimination

Procedure $\mathcal{ALC}$-$\mathtt{Elim}(C_0, \mathcal{T})$:

input : $\mathcal{ALC}$-Konzept C_0, $\mathcal{ALC}$-TBox $\mathcal{T}$
output: „erfüllbar“ / „unerfüllbar“

Berechne Γ_0: Menge aller Typen für C_0 und $\mathcal{T}$

$i := 0$
repeat
  $i := i + 1$
  $\Gamma_i := \{t \in \Gamma_{i-1} \mid t$ nicht schlecht in $\Gamma_{i-1}\}$
until $\Gamma_i = \Gamma_{i-1}$

if *es gibt* $t \in \Gamma_i$ *mit* $C_0 \in t$ **then return** „erfüllbar“
else return „unerfüllbar“

T 5.1 Forts.

Korrektheit

Lemma 5.5

Wenn $\mathcal{ALC}$-$\mathtt{Elim}(C_0, \mathcal{T})$ „erfüllbar“ zurückgibt, dann ist C_0 bezüglich $\mathcal{T}$ erfüllbar.

Beweis. Wenn der Algorithmus „erfüllbar“ zurückgibt, dann gibt es in der resultierenden Typmenge Γ_i einen Typen $t_0 \in \Gamma_i$ mit $C_0 \in t_0$.
Aus Γ_i konstruieren wir Interpretation $\mathcal{I}$ wie folgt.

$\Delta^\mathcal{I} = \Gamma_i$

$A^\mathcal{I} = \{t \mid A \in t\}$ für alle Konzeptnamen A

$r^\mathcal{I} = \{(t, t') \mid \forall r.C \in t$ impliziert $C \in t'\}$ f. a. Rollennamen r

T 5.2

Die Erfüllbarkeit von C_0 bzgl. $\mathcal{T}$ folgt dann aus:

Behauptung. Für alle $C \in \mathrm{sub}(C_0, \mathcal{T})$ und alle $t \in \Gamma_i$ gilt:

$$C \in t \quad \text{impliziert} \quad t \in C^\mathcal{I}$$

T 5.2 Forts.