

Tafelmitschriften zur Vorlesung „Beschreibungslogik“ im Sommersemester 2019

Dr. Jean Christoph Jung, Prof. Dr. Thomas Schneider
AG Theorie der Künstlichen Intelligenz
Fachbereich 3



Stand: 13. Mai 2020

Dieses Dokument ist noch unvollständig und wird regelmäßig aktualisiert.

Inhaltsverzeichnis

2	Grundlagen	3
3	Ausdrucksstärke und Modellkonstruktionen	11
4	Tableau-Algorithmen	18
5	Komplexität	30
6	Effiziente Beschreibungslogiken	31
7	ABoxen und Anfragebeantwortung	32
	Literaturverzeichnis	33

Teil 2

Grundlagen

T2.1 Beispiele für \mathcal{ALC} -Konzepte

Mit den Konzeptnamen

Student, Naturwissenschaft, Professor, Emeritus, PflichtVL, VL, Einfach, Interessant, A, B

und den Rollennamen

studiert, hält, hatÜbungsaufgabe,
 r

kann man z. B. folgende zusammengesetzte \mathcal{ALC} -Konzepte bilden:

- $\text{Student} \sqcap \exists \text{studiert.Naturwissenschaft}$
(beschreibt Studierende, die mindestens eine Naturwissenschaft studieren)
- $\text{Professor} \sqcap \text{Emeritus} \sqcap \forall \text{hält.} \neg \text{PflichtVL}$
(beschreibt Professor*innen im Ruhestand, die keine Pflichtvorlesungen halten)
- $\text{VL} \sqcap \neg \text{PflichtVL} \sqcap \forall \text{hatÜbungsaufgabe.} (\text{Einfach} \sqcup \text{Interessant})$
(beschreibt Wahlvorlesungen, bei denen alle Übungsaufgaben einfach oder interessant sind)
- $A \sqcap \exists r. (\neg B \sqcup \forall r. A)$

(Die Beschreibungen in Klammern werden eigentlich erst richtig klar, wenn die Semantik definiert ist.)

T2.2 Beispiele für Interpretationen und Extensionen

Wir betrachten die Interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ mit

$$\begin{aligned}\Delta^{\mathcal{I}} &= \{s_1, s_2, s_3, v_1, v_2\} \\ \text{Mensch}^{\mathcal{I}} &= \{s_1, s_2, s_3\} \\ \text{Student}^{\mathcal{I}} &= \{s_1, s_2, s_3\} \\ \text{Vorlesung}^{\mathcal{I}} &= \{v_1, v_2\} \\ \text{PflichtVL}^{\mathcal{I}} &= \{v_1\} \\ \text{WahlVL}^{\mathcal{I}} &= \{v_2\} \\ \text{hört}^{\mathcal{I}} &= \{(s_1, v_1), (s_2, v_1), (s_2, v_2), (s_3, v_1)\} \\ \text{bekanntMit}^{\mathcal{I}} &= \{(s_1, s_2), (s_2, s_1), (s_1, s_1), (s_2, s_2), (s_3, s_3)\}.\end{aligned}$$

Jede Interpretation kann in offensichtlicher Weise als (knoten- und kantenbeschrifteter) gerichteter Graph aufgefasst werden; für unsere Beispielinterpretation \mathcal{I} :



Beispiele für die Extensionen einiger zusammengesetzter Konzepte in dieser Interpretation:

$$\begin{aligned}
(\text{VL} \sqcap \text{PflichtVL})^{\mathcal{I}} &= \{v_1, v_2\} \cap \{v_1\} &&= \{v_1\} \\
(\neg \text{VL})^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus \{v_1, v_2\} &&= \{s_1, s_2, s_3\} \\
(\text{Student} \sqcup \text{VL})^{\mathcal{I}} &= \{s_1, s_2, s_3\} \cup \{v_1, v_2\} = \Delta^{\mathcal{I}} \\
(\exists \text{bkenntMit. Student})^{\mathcal{I}} &= \{s_1, s_2, s_3\} \\
(\exists \text{bkenntMit.} \exists \text{bkenntMit. Student})^{\mathcal{I}} &= \{s_1, s_2, s_3\} \\
(\forall \text{hört. PflichtVL})^{\mathcal{I}} &= \{s_1, s_3, v_1, v_2\}
\end{aligned}$$

In der letzten Zeile beachte man die Besonderheit der Werterestriktion (\forall), dass ein Domänenelement d , welches *keine* ausgehenden r -Kanten besitzt, immer eine Instanz von $\forall r.C$ ist, für jedes beliebige Konzept C .

T2.3 Semantik von \top und \perp

Es gelten:

$$\begin{aligned}
\top^{\mathcal{I}} &= (A \sqcup \neg A)^{\mathcal{I}} = A^{\mathcal{I}} \cup (\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}) = \Delta^{\mathcal{I}} \\
\perp^{\mathcal{I}} &= (A \sqcap \neg A)^{\mathcal{I}} = A^{\mathcal{I}} \cap (\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}) = \emptyset^{\mathcal{I}}
\end{aligned}$$

Dabei folgt die erste Gleichheit jeder Zeile aus der Definition von \top bzw. \perp auf Folie 2.9, die zweite Gleichheit aus der Semantik (Def. 2.2) und die dritte aus der Mengenlehre.

T2.4 Beispiele für „unerfüllbar“ und „subsumiert“

- (a) Das Konzept $C = \exists r.A \sqcap \forall r.\neg A$ ist *nicht* erfüllbar:

Angenommen, C sei erfüllbar, d. h. es gibt eine Interpretation \mathcal{I} mit $C^{\mathcal{I}} \neq \emptyset$. Sei $d \in C^{\mathcal{I}}$. Wegen $d \in (\exists r.A)^{\mathcal{I}}$ gibt es ein Element $e \in A^{\mathcal{I}}$ mit $(d, e) \in r^{\mathcal{I}}$. Wegen $d \in (\forall r.\neg A)^{\mathcal{I}}$ gilt aber $e \in (\neg A)^{\mathcal{I}}$, also $e \notin A^{\mathcal{I}}$, was ein Widerspruch zu $e \in A^{\mathcal{I}}$ ist. Also ist die Annahme falsch.

- (b) $\exists r.(A \sqcap B) \sqsubseteq \exists r.A \sqcap \exists r.B$:

Sei \mathcal{I} eine Interpretation und $d \in (\exists r.(A \sqcap B))^{\mathcal{I}}$. Dann gibt es ein Element $e \in (A \sqcap B)^{\mathcal{I}}$ mit $(d, e) \in r^{\mathcal{I}}$. Wegen $e \in A^{\mathcal{I}}$ gilt $d \in (\exists r.A)^{\mathcal{I}}$; wegen $e \in B^{\mathcal{I}}$ gilt $d \in (\exists r.B)^{\mathcal{I}}$. Also ist $d \in (\exists r.A \sqcap \exists r.B)^{\mathcal{I}}$.

Die Rückrichtung dieser Subsumtion gilt nicht – finde ein Gegenbeispiel, d. h. eine Interpretation \mathcal{I} mit $(\exists r.A \sqcap \exists r.B)^{\mathcal{I}} \not\subseteq (\exists r.(A \sqcap B))^{\mathcal{I}}$.

T2.5 Beispiele für TBoxen und deren Semantik

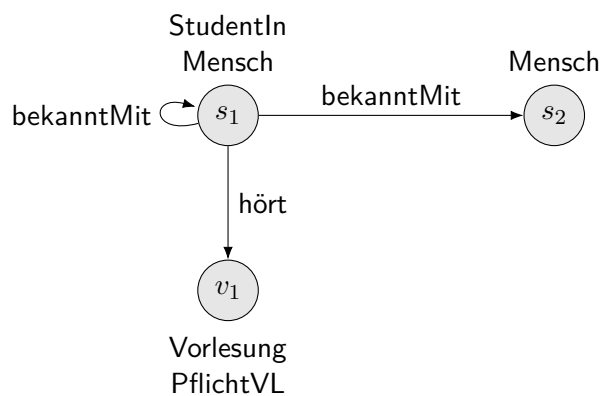
Wir betrachten folgende TBox.

$$\mathcal{T} = \left\{ \begin{array}{ll} \text{StudentIn} & \equiv \text{Mensch} \sqcap \exists \text{hört.Vorlesung} \\ \text{Vorlesung} & \equiv \text{PflichtVL} \sqcup \text{WahlVL} \\ \text{StudentIn} \sqcap \exists \text{hört.Vorlesung} & \sqsubseteq \exists \text{bekanntMit.StudentIn} \\ \text{PflichtVL} \sqcap \text{WahlVL} & \sqsubseteq \perp \end{array} \right\}$$

Die Interpretation aus T2.2 ist Modell von \mathcal{T} . Sie erfüllt z. B. auch die folgende Konzeptinklusion.

$$\text{StudentIn} \sqsubseteq \exists \text{bekanntMit.Mensch} \quad (1)$$

Ein weiteres Modell ist z. B. folgende Interpretation \mathcal{J} .



\mathcal{J} erfüllt ebenfalls die Konzeptinklusion (1) sowie z. B. $\text{VL} \equiv \text{PflichtVL}$.

T2.6 Beispiele für „erfüllbar“ und „subsumiert“ bzgl. TBoxen

Sei \mathcal{T} die TBox aus dem vorangehenden Beispiel.

- (a) Das Konzept

$$C = \text{StudentIn} \sqcap \forall \text{hört.PflichtVL}$$

ist erfüllbar bezüglich \mathcal{T} , denn folgende Interpretation \mathcal{I}' ist ein Modell von \mathcal{T} mit $s_1 \in C^{\mathcal{I}'}$:



Ebenso ist die Interpretation \mathcal{I} aus T2.2 ein Modell von \mathcal{T} mit $s_1 \in C^{\mathcal{I}}$.

- (b) Das Konzept

$$C = \text{StudentIn} \sqcap \forall \text{hört.PflichtVL} \sqcap \exists \text{hört.WahlVL}$$

ist *unerfüllbar* bezüglich \mathcal{T} : Angenommen, C sei erfüllbar bzgl. \mathcal{T} . Dann gibt es ein Modell \mathcal{I} von \mathcal{T} mit einer Instanz $d \in C^{\mathcal{I}}$. Nach der Semantik von „ \sqcap “ (Def. 2.2) gelten (i) $d \in (\forall \text{hört.PflichtVL})^{\mathcal{I}}$ und (ii) $d \in (\exists \text{hört.WahlVL})^{\mathcal{I}}$. Wegen (ii) gibt es ein Element $e \in \text{WahlVL}^{\mathcal{I}}$ mit $(d, e) \in \text{hört}^{\mathcal{I}}$. Wegen (i) ist dann auch $e \in \text{PflichtVL}^{\mathcal{I}}$, also $e \in (\text{PflichtVL} \sqcap \text{WahlVL})^{\mathcal{I}}$. Weil \mathcal{I} jedoch ein Modell von \mathcal{T} ist, kann es wegen der Konzeptinklusion $\text{PflichtVL} \sqcap \text{WahlVL} \sqsubseteq \perp$ aus \mathcal{T} ein solches Element e nicht geben; ein Widerspruch. Also ist die Annahme falsch.

- (c) Für die Konzepte

$$C = \text{StudentIn} \quad \text{und} \quad D = \exists \text{bekanntMit.StudentIn}$$

gilt $\mathcal{T} \models C \sqsubseteq D$: Sei \mathcal{I} ein Modell von \mathcal{T} und $d \in C^{\mathcal{I}}$, d. h. $d \in \text{StudentIn}^{\mathcal{I}}$. Zu zeigen ist $d \in D^{\mathcal{I}}$, d. h. $d \in (\exists \text{bekanntMit.StudentIn})^{\mathcal{I}}$.

Wegen der ersten Zeile von \mathcal{T} gilt $d \in (\exists \text{hört.Vorlesung})^{\mathcal{I}}$, also auch $d \in (\text{StudentIn} \sqcap \exists \text{hört.Vorlesung})^{\mathcal{I}}$. Mit Zeile 3 von \mathcal{T} folgt dann $d \in (\exists \text{bekanntMit.StudentIn})^{\mathcal{I}}$.

Dies ist bereits Schlussfolgern, denn wir haben implizites Wissen aus \mathcal{T} abgeleitet:

- (a) Es *kann* Student*innen geben, die nur Pflichtvorlesungen hören.
- (b) Es kann *keine* Student*innen geben, die nur Pflichtvorlesungen, aber mindestens eine Wahlvorlesung hören.
- (c) Jede*r Student*in ist mit mindestens einer/m Student*in bekannt.

T2.7 Beweis der Monotonie von \mathcal{ALC} (Lemma 2.7)

Lemma 2.7 Seien \mathcal{T}_1 und \mathcal{T}_2 TBoxen mit $\mathcal{T}_1 \subseteq \mathcal{T}_2$. Dann gilt:

- (1) Wenn C erfüllbar bezüglich \mathcal{T}_2 ist, dann ist C erfüllbar bezüglich \mathcal{T}_1 .
- (2) Wenn $\mathcal{T}_1 \models C \sqsubseteq D$, dann $\mathcal{T}_2 \models C \sqsubseteq D$.

Beweis.

- (1) Sei C erfüllbar bezüglich \mathcal{T}_2 . Dann gibt es ein Modell \mathcal{I} von \mathcal{T}_2 mit $C^{\mathcal{I}} \neq \emptyset$. Da \mathcal{I} Modell von \mathcal{T}_2 ist, erfüllt \mathcal{I} alle Konzeptinklusionen in \mathcal{T}_2 , also wegen $\mathcal{T}_1 \subseteq \mathcal{T}_2$ auch alle Konzeptinklusionen in \mathcal{T}_1 , und somit ist \mathcal{I} auch Modell von \mathcal{T}_1 . Also gibt es ein Modell \mathcal{I} von \mathcal{T}_1 mit $C^{\mathcal{I}} \neq \emptyset$; d. h. C ist erfüllbar bezüglich \mathcal{T}_1 .
- (2) Wir beweisen die Kontraposition. Es gelte $\mathcal{T}_2 \not\models C \sqsubseteq D$.¹ Dann gibt es ein Modell \mathcal{I} von \mathcal{T}_2 mit $C^{\mathcal{I}} \not\subseteq D^{\mathcal{I}}$. Wie in (1) ist \mathcal{I} auch Modell von \mathcal{T}_1 , also $\mathcal{T}_1 \not\models C \sqsubseteq D$. \square

Auf der Folie steht auch: „Die Umkehrungen von (1) und (2) sind im Allgemeinen *nicht* richtig.“ Davon kann man sich mittels einfacher Gegenbeispiele überzeugen: z. B. ist mit $\mathcal{T}_1 = \emptyset$ und $\mathcal{T}_2 = \{A \sqsubseteq B\}$ die Umkehrung von (2) widerlegt, denn $\mathcal{T}_2 \models A \sqsubseteq B$, aber $\mathcal{T}_1 \not\models A \sqsubseteq B$.

T2.8 Beispiel für Subsumtion als Ordnungsrelation

Wir betrachten folgende TBox.

$$\mathcal{T} = \left\{ \begin{array}{l} \text{PC} \sqsubseteq \text{Gerät} \sqcap \exists \text{hatTeil.CPU} \\ \text{PC} \equiv \text{Desktop} \sqcup \text{Laptop} \\ \text{Desktop} \sqsubseteq \neg \text{Laptop} \\ \text{MobilerPC} \equiv \text{PC} \sqcap \neg \text{Desktop} \end{array} \right\}$$

Die dritte Zeile von \mathcal{T} ist äquivalent zu $\text{Desktop} \sqcap \text{Laptop} \sqsubseteq \perp$, wie man leicht zeigt (probiere es selbst aus).

Die Ordnung „ \sqsubseteq bezüglich \mathcal{T} “ kann man durch folgendes Hasse-Diagramm darstellen:



Dass die Relation „ \sqsubseteq bezüglich \mathcal{T} “ *nicht* antisymmetrisch ist, zeigt sich in diesem Beispiel dadurch, dass der Knoten unten rechts zwei Beschriftungen hat, also $\mathcal{T} \models \text{Laptop} \equiv \text{MobilerPC}$. Wäre die Relation antisymmetrisch, dann dürfte nicht gleichzeitig $\mathcal{T} \models \text{Laptop} \sqsubseteq \text{MobilerPC}$ und $\mathcal{T} \models \text{MobilerPC} \sqsubseteq \text{Laptop}$ gelten.

¹Das Zeichen $\not\models$ steht für „nicht \models “, also bedeutet $\mathcal{T} \not\models C \sqsubseteq D$, dass die Beziehung $\mathcal{T} \models C \sqsubseteq D$ *nicht* gilt (d. h. bezüglich \mathcal{T} wird C *nicht* von D subsumiert).

T2.9 Wechselseitige Reduktion der Schlussfolgerungsprobleme

Lemma 2.9

- (1) Subsumtion ist polynomiell reduzierbar auf (Un)erfüllbarkeit:
 $\mathcal{T} \models C \sqsubseteq D$ gdw. $C \sqcap \neg D$ unerfüllbar bezüglich \mathcal{T}
- (2) Erfüllbarkeit ist polynomiell reduzierbar auf (Nicht-)Äquivalenz:
 C erfüllbar bezüglich \mathcal{T} gdw. $\mathcal{T} \not\models C \equiv \perp$
- (3) Äquivalenz ist polynomiell reduzierbar auf Subsumtion:
 $\mathcal{T} \models C \equiv D$ gdw. $\mathcal{T} \models \top \sqsubseteq (C \sqcap D) \sqcup (\neg C \sqcap \neg D)$

Beweis. Wir beweisen exemplarisch Punkt (1). Die Beweise der anderen zwei Punkte sind analog.

$$\begin{aligned}
 \mathcal{T} \models C \sqsubseteq D & \text{ gdw. } \text{für alle Modelle } \mathcal{I} \text{ von } \mathcal{T} \text{ gilt } C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \\
 & \text{gdw. } \text{für alle Modelle } \mathcal{I} \text{ von } \mathcal{T} \text{ gilt } C^{\mathcal{I}} \cap (\Delta^{\mathcal{I}} \setminus D^{\mathcal{I}}) = \emptyset \\
 & \text{gdw. } \text{für alle Modelle } \mathcal{I} \text{ von } \mathcal{T} \text{ gilt } (C \sqcap \neg D)^{\mathcal{I}} = \emptyset \\
 & \text{gdw. } C \sqcap \neg D \text{ unerfüllbar bezüglich } \mathcal{T}
 \end{aligned}$$

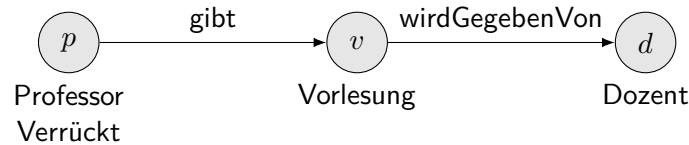
Hier gilt das erste „gdw.“ wegen der Definition von Subsumtion (Def. 2.6), und die zweite Zeile ist äquivalent zur ersten, weil für beliebige Mengen M_1, M_2 gilt, dass $M_1 \subseteq M_2$ gdw. $M_1 \cap \bar{M}_2 = \emptyset$, wobei \bar{M}_2 das Komplement von M_2 ist. Man überzeuge sich davon anhand eines Venn-Diagramms. Die dritte Zeile ist schließlich äquivalent zur dritten wegen der Semantik von \sqcap und \neg (Def. 2.2), und die vierte ist äquivalent dazu wegen der Definition von Unerfüllbarkeit (Def. 2.6). \square

T2.10 Beispiel für inverse Rollen

Betrachte folgende \mathcal{ALC} -TBox:

$$\mathcal{T} = \{ \text{Professor} \sqsubseteq \text{Verrückt} \sqcap \exists \text{gibt.Vorlesung} \\
 \text{Vorlesung} \sqsubseteq \forall \text{wirdGegebenVon}.\neg \text{Verrückt} \}$$

Intuitiv sollte **Professor** unerfüllbar bezüglich \mathcal{T} sein; dies ist aber nicht der Fall, weil **Professor** in folgendem Modell von \mathcal{T} eine Instanz hat:



In \mathcal{ALCI} kann man die zweite Konzeptinklusion in \mathcal{T} durch

$$\text{Vorlesung} \sqsubseteq \forall \text{gibt}^-. \neg \text{Verrückt}$$

ersetzen. Dann wird **Professor** unerfüllbar bezüglich \mathcal{T} (aber \mathcal{T} hat immer noch Modelle).

T2.11 Beispiele für Zahlenrestriktionen

Definition einer Hand als ein Organ mit genau fünf Fingern:

$$\text{Hand} \equiv \text{Organ} \sqcap (\geq 5 \text{ hatTeil.Finger}) \sqcap (\leq 5 \text{ hatTeil.Finger})$$

Forderung, dass in jedem Semester mindestens zwei Wahlpflichtmodule angeboten werden:

$$\text{Semester} \sqsubseteq (\geq 2 \text{ angeboten.Wahlpflichtmodul})$$

T2.12 Beispiele für die Übersetzung von \mathcal{ALC} nach FO

Wir betrachten folgendes \mathcal{ALC} -Konzept.

$$C = \text{Mensch} \sqcap \exists \text{hört.Vorlesung}$$

Wir erhalten $\pi_x(C)$ anhand der induktiven Definition auf der Folie, indem wir mit den Konzeptnamen beginnen und uns zu komplexeren Konzepten „hocharbeiten“:

$$\pi_x(\text{Mensch}) = \text{Mensch}(x)$$

$$\pi_y(\text{Vorlesung}) = \text{Vorlesung}(y)$$

$$\pi_x(\exists \text{hört.Vorlesung}) = \exists y(\text{hört}(x, y) \wedge \text{Vorlesung}(y))$$

$$\pi_x(\text{Mensch} \sqcap \exists \text{hört.Vorlesung}) = \text{Mensch}(x) \wedge \exists y(\text{hört}(x, y) \wedge \text{Vorlesung}(y))$$

$$\text{also} \quad \pi_x(C) = \text{Mensch}(x) \wedge \exists y(\text{hört}(x, y) \wedge \text{Vorlesung}(y))$$

Wenn wir außerdem die TBox

$$\mathcal{T} = \{\text{StudentIn} \equiv \text{Mensch} \sqcap \exists \text{hört.Vorlesung}\}$$

gegeben haben, dann ist

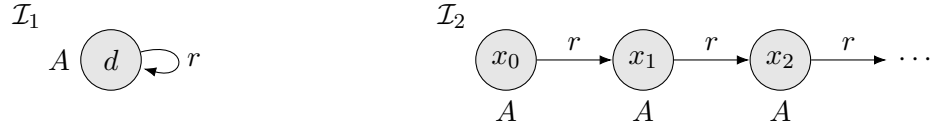
$$\pi(\mathcal{T}) = \forall x (\text{StudentIn}(x) \leftrightarrow \text{Mensch}(x) \wedge \exists y(\text{hört}(x, y) \wedge \text{Vorlesung}(y))).$$

Teil 3

Ausdrucksstärke und Modellkonstruktionen

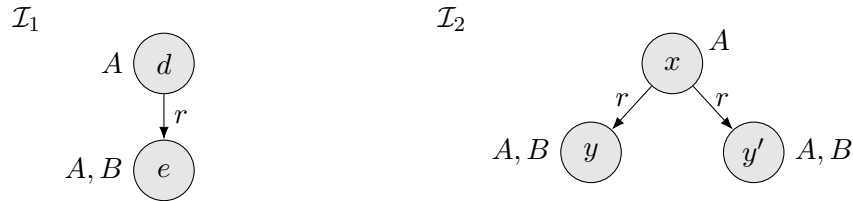
T3.1 Beispiele für Bisimulationen

1. Für die Interpretationen



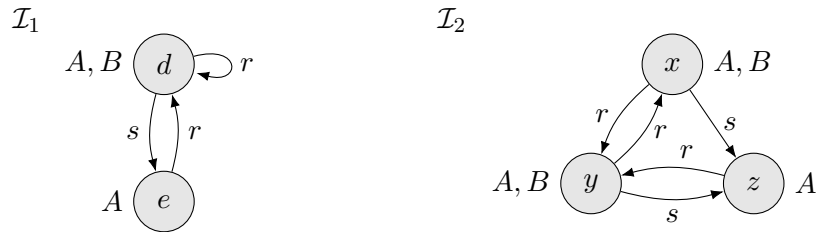
ist $\rho = \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ eine Bisimulation.

2. Für die Interpretationen



ist $\rho = \{(d, x), (e, y), (e, y')\}$ eine Bisimulation.

3. Für die Interpretationen



ist $\rho = \{(d, x), (d, y), (e, z)\}$ eine Bisimulation.

T3.2 Beweis des Bisimulationstheorems

Theorem 3.2 Seien $\mathcal{I}_1, \mathcal{I}_2$ Interpretationen, $d_1 \in \Delta^{\mathcal{I}_1}$ und $d_2 \in \Delta^{\mathcal{I}_2}$.

Wenn $(\mathcal{I}_1, d_1) \sim (\mathcal{I}_2, d_2)$, dann gilt für alle \mathcal{ALC} -Konzepte C :

$$d_1 \in C^{\mathcal{I}_1} \quad \text{gdw.} \quad d_2 \in C^{\mathcal{I}_2}$$

Beweis. Sei ρ eine Bisimulation zwischen \mathcal{I}_1 und \mathcal{I}_2 mit $d_1 \rho d_2$. Wir beweisen die Behauptung per Induktion über die Struktur von C .

Induktionsanfang. Hier ist $C = A$ für einen Konzeptnamen A . Nach Bedingung (1) für Bisimulationen (Definition 3.1) gilt wie gewünscht: $d_1 \in A^{\mathcal{I}_1}$ gdw. $d_2 \in A^{\mathcal{I}_2}$

Induktionsschritt. Wir müssen fünf Fälle gemäß des äußersten Konstruktors von C unterscheiden ($\neg, \sqcap, \sqcup, \exists, \forall$). Wegen der (leicht nachzuweisenden) Äquivalenzen

$$C \sqcup D \equiv \neg(\neg C \sqcap \neg D) \quad \text{und} \quad \forall r.C \equiv \neg \exists r.\neg C$$

genügt es, wenn wir uns auf die drei Fälle \neg, \sqcap, \exists beschränken.

$$C = \neg D$$

$$\begin{array}{lll} d_1 \in C^{\mathcal{I}_1} & \text{gdw.} & d_1 \notin D^{\mathcal{I}_1} & (\text{Semantik „}\neg\text{“}) \\ & & d_2 \notin D^{\mathcal{I}_2} & (\text{Induktionsvoraussetzung}) \\ & & d_2 \in C^{\mathcal{I}_2} & (\text{Semantik „}\neg\text{“}) \end{array}$$

$$C = D \sqcap E$$

$$\begin{array}{lll} d_1 \in C^{\mathcal{I}_1} & \text{gdw.} & d_1 \in D^{\mathcal{I}_1} \text{ und } d_1 \in E^{\mathcal{I}_1} & (\text{Semantik „}\sqcap\text{“}) \\ & & d_2 \in D^{\mathcal{I}_2} \text{ und } d_2 \in E^{\mathcal{I}_2} & (\text{Induktionsvoraussetzung}) \\ & & d_2 \in C^{\mathcal{I}_2} & (\text{Semantik „}\sqcap\text{“}) \end{array}$$

$$C = \exists r.D$$

Für die Richtung „ \Rightarrow “ argumentieren wir so:

$$\begin{array}{ll} d_1 \in C^{\mathcal{I}_1} & \\ \Rightarrow \text{es gibt } e_1 \in \Delta^{\mathcal{I}_1} \text{ mit } (d_1, e_1) \in r^{\mathcal{I}_1} \text{ und } e_1 \in D^{\mathcal{I}_1} & (\text{Semantik „}\exists\text{“}) \\ \Rightarrow \text{es gibt } e_2 \in \Delta^{\mathcal{I}_2} \text{ mit } (d_2, e_2) \in r^{\mathcal{I}_2} \text{ und } e_1 \rho e_2 & (\text{Bedingung (2) Bisim.}) \\ \Rightarrow e_2 \in D^{\mathcal{I}_2} & (\text{Induktionsvoraus.}) \\ \Rightarrow d_2 \in (\exists r.D)^{\mathcal{I}_2} & (\text{Semantik „}\exists\text{“}) \end{array}$$

Das Argument für die Richtung „ \Leftarrow “ ist analog, unter Verwendung von Bedingung (3) für Bisimulationen. \square

T3.3 Nichtausdrückbarkeit konkreter Eigenschaften

Theorem 3.4 In \mathcal{ALC} sind *nicht ausdrückbar*:

- das \mathcal{ALCI} -Konzept $\exists r^{\neg}.\top$
- die \mathcal{ALCQ} -Konzepte
 - $(\leq n r.\top)$, für alle $n > 0$
 - $(\geq n r.\top)$, für alle $n > 1$

Beweis. Siehe Proposition 3.3 und 3.4 in [BHLS17].

T3.4 Gegenbeispiel für Rückrichtung Bisimulationstheorem

Behauptung. Es gibt Interpretationen \mathcal{I} und \mathcal{J} und $d \in \mathcal{I}$, $e \in \mathcal{J}$, so dass

- (i) $d \in C^{\mathcal{I}}$ gdw. $e \in C^{\mathcal{J}}$ für alle \mathcal{ALC} -Konzepte C ,
- (ii) aber $(\mathcal{I}, d) \not\sim (\mathcal{J}, e)$.

Beweis. Betrachte die folgenden Interpretationen \mathcal{I} und \mathcal{J} .

... Bild folgt ...

Es gilt (ii): versucht man, eine Bisimulation ρ mit $d \rho e$ zu konstruieren, so benötigt man wegen $(e, e') \in r^{\mathcal{I}}$ einen r -Nachfolger d' von d mit $d' \rho e'$. Da jeder r -Nachfolger von d aber nur endlich viele weitere Nachfolger hat, kann man wegen des unendlichen r -Pfads unterhalb von e' irgendwann nicht mehr Bedingung (3) von Bisimulationen gewährleisten.

Außerdem kann man mittels struktureller Induktion zeigen, dass (i) gilt. □

T3.5 Beispiel für ein Baummodell

Sei $C = A \sqcap \exists s.B \sqcap \forall s.\exists r.A$ und $\mathcal{T} = \{\top \sqsubseteq \exists s.A\}$. Ein Baummodell von C und \mathcal{T} :



T3.6 Beispiel für das Unravelling

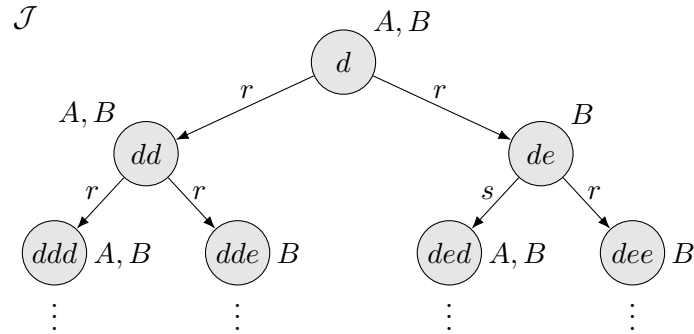
Wir betrachten folgende Interpretation \mathcal{I} .



Dann gibt es beispielsweise folgende d -Pfade:

- $\rho = ddedee$ mit $\text{end}(\rho) = e$
- $\rho' = deeed$ mit $\text{end}(\rho') = d$

Das Unravelling von \mathcal{I} an Stelle d gemäß Definition 3.7 ist folgende Interpretation \mathcal{J} .



T3.7 Beweis des Unravelling-Lemmas

Lemma 3.8 Für alle \mathcal{ALC} -Konzepte C und alle $p \in \Delta^{\mathcal{J}}$ gilt:

$$\text{end}(p) \in C^{\mathcal{I}} \quad \text{gdw.} \quad p \in C^{\mathcal{J}}$$

Beweis. Mit dem Bisimulationstheorem (Theorem 3.2) genügt es zu zeigen, dass $\text{end}(p)$ und p bisimilar sind, d. h. $(\mathcal{I}, \text{end}(p)) \sim (\mathcal{J}, p)$. Siehe dazu Lemma 3.22 in [BHLS17].

T3.8 Beweis der Baummodelleigenschaft von \mathcal{ALC}

Theorem 3.6 Wenn ein \mathcal{ALC} -Konzept C bezüglich einer \mathcal{ALC} -TBox \mathcal{T} erfüllbar ist, dann haben C und \mathcal{T} ein gemeinsames Baummodell \mathcal{I} .

Beweis. Siehe Theorem 3.24 in [BHLS17].

T3.9 Beispiel für Teilkonzepte

Sei $C = \forall r. \exists r. (A \sqcap B)$. Dann ist $\text{sub}(C) = \{A, B, A \sqcap B, \exists r. (A \sqcap B), \forall r. \exists r. (A \sqcap B)\}$.

Sei $\mathcal{T} = \{A \sqsubseteq \exists r. B, \forall r. B \sqsubseteq A\}$. Dann ist $\text{sub}(\mathcal{T}) = \{A, B, \exists r. B, \forall r. B\}$.

T3.10 Beispiel für Typen und Filtration

Seien $C = A \sqcap B$ und $\mathcal{T} = \{A \sqsubseteq \exists r.A\}$. Dann ist $\text{sub}(C, \mathcal{T}) = \{A, B, A \sqcap B, \exists r.A\}$.

Wir betrachten die folgende Interpretation \mathcal{I} .



Dann gilt für d und die vier Teilkonzepte in $\text{sub}(C, \mathcal{T})$: $d \in A^{\mathcal{I}}$, $d \notin B^{\mathcal{I}}$, $d \notin (A \sqcap B)^{\mathcal{I}}$ und $d \in (\exists r.A)^{\mathcal{I}}$. Also ist

$$t_{\mathcal{I}}(d) = \{A, \exists r.A\}.$$

Analog erhält man:

$$t_{\mathcal{I}}(e) = \{A, \exists r.A\} \quad t_{\mathcal{I}}(f) = \{A, B, A \sqcap B, \exists r.A\} \quad t_{\mathcal{I}}(g) = \{A, \exists r.A\}$$

Es gilt also $d \simeq e \simeq g \not\simeq f$, und somit gibt es zwei Äquivalenzklassen bezüglich \simeq :

$$[d] = \{d, e, g\} \quad [f] = \{f\}$$

Die Filtration von \mathcal{I} bezüglich C und \mathcal{T} gemäß Definition 3.17 ist dann folgende Interpretation \mathcal{J} .



T3.11 Beweis des Filtrationstheorems

Theorem 3.17 Sei \mathcal{I} ein Modell von C und \mathcal{T} , und sei \mathcal{J} die Filtration von \mathcal{I} bezüglich C und \mathcal{T} . Dann ist auch \mathcal{J} ein Modell von C und \mathcal{T} .

Beweis. Wir verwenden die folgende Hilfsaussage.

$$\text{Für alle } d \in \Delta^{\mathcal{I}} \text{ und } D \in \text{sub}(C, \mathcal{T}) \text{ gilt: } d \in D^{\mathcal{I}} \quad \text{gdw.} \quad [d] \in D^{\mathcal{J}}$$

Für den Beweis dieser Hilfsaussage siehe Lemma 3.15 in [BHLS17].

Da \mathcal{I} ein Modell von C und \mathcal{T} ist, gibt es ein Element $d \in C^{\mathcal{I}}$. Mit der Hilfsaussage folgt $[d] \in C^{\mathcal{J}}$; somit ist \mathcal{J} ein Modell von C .

Um zu zeigen, dass \mathcal{J} auch ein Modell von \mathcal{T} ist, betrachten wir eine beliebige Konzeptinklusion $D \sqsubseteq E \in \mathcal{T}$ und eine beliebige Instanz $[d] \in D^{\mathcal{J}}$. Mit der Hilfsaussage folgt $d \in D^{\mathcal{I}}$; also $d \in E^{\mathcal{I}}$ (da \mathcal{T} Modell von \mathcal{I} ist); also mit der Hilfsaussage $[d] \in E^{\mathcal{J}}$. \square

T3.12 \mathcal{ALCQI} hat nicht die endliche Modelleigenschaft

Betrachte die TBox $\mathcal{T} = \{(1) \top \sqsubseteq \exists r. \neg A, (2) \top \sqsubseteq (\leq 1 r^-. \top)\}$. Dann hat der Konzeptname A bezüglich \mathcal{T} nur *unendliche Modelle*:

Sei \mathcal{I} ein Modell von A und \mathcal{T} und $d_0 \in \Delta^{\mathcal{I}}$. Wir müssen zeigen, dass $|\Delta^{\mathcal{I}}| = \infty$.

Da \mathcal{I} Modell von \mathcal{T} ist, muss es wegen (1) einen r -Nachfolger d_1 von d_0 geben mit $d_1 \in (\neg A)^{\mathcal{I}}$. Da $d_0 \in A^{\mathcal{I}}$, muss $d_1 \neq d_0$ gelten.

Wegen (1) muss es wiederum einen r -Nachfolger d_2 von d_1 geben mit $d_2 \in (\neg A)^{\mathcal{I}}$. Wie im vorigen Fall muss $d_1 \neq d_0$ gelten. Außerdem muss $d_2 \neq d_1$ gelten, da sonst d_1 zwei r -Vorgänger hätte (d_0 und d_1) und dann (2) verletzt wäre.

Dieses Argument kann man so fortsetzen und immer wieder die Existenz eines neuen Elements d_{i+1} folgern, dass verschieden von allen d_0, \dots, d_i sein muss. Deshalb muss $\Delta^{\mathcal{I}}$ unendlich sein.

T3.13 Anzahl der Interpretationen der Größe $\leq 2^n$

Behauptung. Sei $n = |C| + |\mathcal{T}|$. Dann gibt es höchstens $2^{2^{5n}}$ Interpretationen \mathcal{I} mit $|\Delta^{\mathcal{I}}| \leq 2^n$.

Beweis. Zunächst betrachten wir die Anzahl aller Interpretationen \mathcal{I} mit $|\Delta^{\mathcal{I}}| = 2^n$.

Jedes Element $d \in \Delta^{\mathcal{I}}$ kann in $A^{\mathcal{I}}$ sein oder nicht, für jeden der $\leq n$ Konzeptnamen A in $\text{sub}(C, \mathcal{T})$. Also gibt es $2^{2^n \cdot n}$ Möglichkeiten für die Extensionen der Konzeptnamen in C und \mathcal{T} .

Jedes Paar von Elementen $(d, e) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ kann in $r^{\mathcal{I}}$ sein oder nicht, für jeden der $\leq n$ Rollennamen r in $\text{sub}(C, \mathcal{T})$. Also gibt es $2^{2^n \cdot 2^n \cdot n}$ Möglichkeiten für die Extensionen der Rollennamen in C und \mathcal{T} .

Die Gesamtzahl der möglichen Extensionen aller Konzept- und Rollennamen in C und \mathcal{T} ist dann das Produkt dieser beiden Zahlen, also:

$$\begin{aligned} 2^{2^n \cdot n} \cdot 2^{2^n \cdot 2^n \cdot n} &= 2^{2^n \cdot n + 2^{2^n} \cdot n} \\ &\leq 2^{2^{2^n} + 2^{3n}} \\ &\leq 2^{2 \cdot 2^{3n}} \\ &= 2^{2^{3n+1}} \\ &\leq 2^{2^{4n}} \end{aligned}$$

Betrachtet man nun die Anzahl der Interpretationen \mathcal{I} mit $|\Delta^{\mathcal{I}}| \leq 2^n$, dann sind dies maximal $2^n \cdot 2^{2^{4n}} = 2^{2^{4n} + n} \leq 2^{2^{5n}}$. \square

Teil 4

Tableau-Algorithmen

T4.1 Umwandlung in NNF

Lemma 4.2. Jedes Konzept kann in Linearzeit in ein äquivalentes Konzept in NNF umgewandelt werden.

Beweis. Dies geschieht durch erschöpfendes Anwenden folgender Regeln.

- Auflösen doppelter Negation: ersetze $\neg\neg C$ durch C
- de Morgan: ersetze $\neg(C \sqcap D)$ durch $\neg C \sqcup \neg D$; ersetze $\neg(C \sqcup D)$ durch $\neg C \sqcap \neg D$
- Dualität von \exists und \forall : ersetze $\neg\exists r.C$ durch $\forall r.\neg C$; ersetze $\neg\forall r.C$ durch $\exists r.\neg C$

Jede Regelanwendung ist äquivalenzerhaltend und schiebt die entsprechende Negation weiter nach innen (oder löst sie auf).

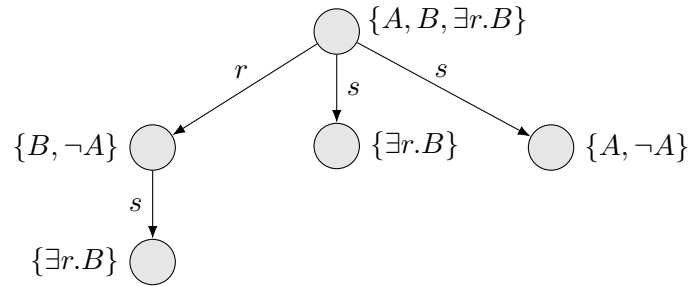
Man kann zeigen, dass linear viele Regelanwendungen ausreichen. \square

T4.2 Beispiel I-Baum

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

Dann ist $\text{sub}(C_0) = \{A, B, \neg A, \exists r.B, \neg A \sqcap \exists r.B, \forall r.(\neg A \sqcap \exists r.B), C_0\}$.

Der folgende Baum ist ein I-Baum für C_0 .



Beachte, dass die Knotenbeschriftung im Allgemeinen nicht der Semantik genügen muss; die Tableau-Regeln werden jedoch dafür sorgen, dass sie das (größtenteils) tut.

T4.3 Beispiel Tableau-Algorithmus

Sei $C_0 = (\exists r.A \sqcap \exists r.\neg A) \sqcap (\forall r.A \sqcup \forall r.B)$. Im Folgenden ist ein möglicher² Lauf des Tableau-Algorithmus auf C_0 angegeben. Nach den ersten drei Schritten liefert dieser folgendes Zwischenergebnis:

	C_0	(1)
\circ	$\exists r.A \sqcap \exists r.\neg A$	(2a)
	$\forall r.A \sqcup \forall r.B$	(2b)
	$\exists r.A, \exists r.\neg A$	(3)

²Da die Reihenfolge der Regelanwendungen nicht festgelegt ist, gibt es im Allgemeinen mehrere Läufe. Das Ergebnis (Menge der vollständigen I-Bäume) ist jedoch nicht von der Reihenfolge abhängig.

Dabei wurden folgende Regeln angewendet.

- (1) initialer Baum \mathcal{B}_{ini}
- (2a), (2b) \sqcap -Regel auf (1)
- (3) \sqcap -Regel auf (2a)

Als nächstes wenden wir die \sqcup -Regel auf $\forall r.A \sqcup \forall r.B$ (2b) an. Dadurch erhalten wir zwei I-Bäume \mathcal{B}_1 und \mathcal{B}_2 :



In \mathcal{B}_1 lässt sich die \exists -Regel auf die beiden Konzepte in (3) anwenden, wodurch jeweils ein neuer r -Nachfolger des Wurzelknotens erzeugt wird:



Nun lässt sich die \forall -Regel auf (4a) und den rechten Nachfolger anwenden (der linke ist bereits mit A beschriftet), und wir erhalten:



Dieser I-Baum ist vollständig (denn es ist keine weitere Regel anwendbar), und er enthält einen offensichtlichen Widerspruch wegen (6) und (7) – ab jetzt immer mit dem Symbol ζ gekennzeichnet.

Schließlich wenden wir in \mathcal{B}_2 dieselben Regeln an und erhalten folgenden I-Baum:



Es wurden also folgende Regeln angewendet:

- (8) \exists -Regel auf $\exists r.A$ (3)
- (9) \exists -Regel auf $\exists r.\neg A$ (3)
- (10) \forall -Regel auf (4b) und linken r -Nachfolger
- (11) \forall -Regel auf (4b) und rechten r -Nachfolger

Dieser Baum ist vollständig, enthält aber *keinen* offensichtlichen Widerspruch. Somit gibt der Algorithmus auf der Eingabe C_0 „erfüllbar“ zurück.

T4.4 Verzweigungsgrad der I-Bäume

Behauptung 1. Es werden nur I-Bäume mit einem Verzweigungsgrad von maximal $|C_0|$ generiert.

Beweis. Nur die \exists -Regel generiert Nachfolgerknoten, und zwar höchstens einen pro Konzept $\exists r.C$ in $\text{sub}(C_0)$. Nach Lemma 3.13 ist aber $|\text{sub}(C_0)| \leq C_0$. \square

T4.5 Tiefe der I-Bäume

Behauptung 2. Es werden nur I-Bäume mit einer Tiefe von maximal $|C_0|$ generiert.

Beweis. Dazu genügt es, folgende Behauptung zu beweisen:

Behauptung 2a. Wenn v ein Knoten mit Tiefe i ist, dann gilt für alle $C \in \mathcal{L}(v)$:

$$\text{rd}(C) \leq \text{rd}(C_0) - i \quad ((*))$$

Dabei zählen wir die Tiefe eines Knotens beginnend von der Wurzel, welche die Tiefe 0 hat. Behauptung 2a besagt also, dass die Rollentiefe der Konzepte in den Knotenbeschriftungen mit der Tiefe eines Knotens abnimmt.

Es ist leicht zu sehen, dass die gewünschte Behauptung 2 bereits aus Behauptung 2a folgt: Wegen der Tableau-Regeln ist jeder Knoten mit mindestens einem Konzept beschriftet

(der initiale Knoten mit C_0 und alle weiteren Knoten mit dem C aus der \exists -Regel). Wenn ein generierter I-Baum also eine Tiefe $k > |C_0|$ hätte, dann gäbe es einen Knoten v der Tiefe k und in dessen Beschriftung $\mathcal{L}(v)$ ein Konzept C . Wegen Behauptung 2a wäre dann aber $\text{rd}(C) < 0$, was nicht möglich ist.

Beweis von Behauptung 2a. Wir verwenden Induktion über die Anzahl der Regelnwendungen.

Induktionsanfang. Nach 0 Regelnwendungen gibt es nur den initialen Knoten v_{ini} mit $\mathcal{L}(v_{\text{ini}}) = \{C_0\}$. Ungleichung (*) folgt, da $i = 0$.

Induktionsschritt. Hier unterscheiden wir vier Fälle nach der Regel, durch deren Anwendung das Konzept C zur Knotenbeschriftung $\mathcal{L}(v)$ hinzugefügt wurde.

\sqcap -Regel. Vor der Anwendung der Regel gab es ein Konzept $C \sqcap D \in \mathcal{L}(v)$, und durch die Anwendung wurden C und D zu $\mathcal{L}(v)$ hinzugefügt. Nach Induktionsvoraussetzung ist $\text{rd}(C \sqcap D) \leq \text{rd}(C_0) - i$, also auch $\text{rd}(C) \leq \text{rd}(C_0) - i$ wegen $C \leq \text{rd}(C \sqcap D)$. Analog für $\text{rd}(D)$.

\sqcup -Regel. Analog (probiert es selbst aus).

\exists -Regel. Vor der Anwendung der Regel gab es einen Vorgängerknoten v' von v mit $\exists r.C \in \mathcal{L}(v')$, und durch die Anwendung wurde (v', r, v) zu E hinzugefügt und $\mathcal{L}(v) = \{C\}$ gesetzt. Nun gilt:

$$\begin{aligned} \text{rd}(C) &= \text{rd}(\exists r.C) - 1 && \text{(Definition rd)} \\ &\leq \left(\text{rd}(C_0) - (i - 1) \right) - 1 && \text{(Induktionsvorauss.; } v' \text{ hat Tiefe } i - 1) \\ &= \text{rd}(C_0) - i \end{aligned}$$

\forall -Regel. Ähnlich (probiert es selbst aus). □

T4.6 Anzahl der Regelnwendungen pro I-Baum

Es bleibt nur noch zu zeigen, dass für $k := |C_0|$ gilt:

$$k^k \cdot k \leq 2^{2k^2}$$

Das kann man wie folgt sehen:

$$\begin{aligned} k^k \cdot k &= k^{k+1} && \text{(Potenzgesetze)} \\ &\leq k^{2k} && \text{(da } k \geq 1) \\ &= (2^{\log k})^{2k} && \text{(Definition Zweierlogarithmus log)} \\ &= 2^{\log k \cdot 2k} && \text{(Potenzgesetze)} \\ &\leq 2^{k \cdot 2k} && \text{(da } \log(k) \leq k) \\ &= 2^{2k^2} \end{aligned}$$

□

T4.7 Letzter Schritt im Terminierungsbeweis

Wir ordnen jeder Menge M_i von I-Bäumen eine Multimenge MM_i wie folgt zu: Für jeden Baum $\mathcal{B} \in M_i$ enthält MM_i die Zahl

$m(\mathcal{B}) = „n \text{ minus die Anzahl } j \text{ der Regelanwendungen, mittels derer } \mathcal{B} \text{ generiert wurde}“.$

Somit ist MM_i eine Multimenge über der Grundmenge \mathbb{N} . Da $<$ auf \mathbb{N} wohlfundiert ist, ist mit Theorem 4.7 auch $<_{mul}$ auf $MM(\mathbb{N})$ wohlfundiert. Außerdem gilt $MM_i >_{mul} MM_{i+1}$ für jedes $i \geq 0$, denn mit jeder Regelanwendung wird in M_i ein I-Baum durch maximal zwei I-Bäume $\mathcal{B}_1, \mathcal{B}_2$ ersetzt mit $m(\mathcal{B}_1), m(\mathcal{B}_2) < m(\mathcal{B})$; somit erhält man MM_{i+1} aus MM_i , indem man $m(\mathcal{B})$ durch die kleineren Zahlen $m(\mathcal{B}_1), m(\mathcal{B}_2)$ ersetzt.

Wegen der Wohlfundiertheit von $<_{mul}$ und der Beobachtung $MM_i >_{mul} MM_{i+1}$ muss die Folge der MM_i endlich sein. \square

T4.8 Beweis Resolutionssatz, Korrektheit

Behauptung. Für alle Konzepte C und Knoten $v \in V$ gilt: $C \in \mathcal{L}(v)$ impliziert $v \in C^{\mathcal{I}}$

Zunächst zeigen wir, dass diese Behauptung die Aussage des Theorems 4.8 impliziert.

Da $C_0 \in \mathcal{L}(v_{ini})$ in \mathcal{B}_{ini} , gilt auch $C_0 \in \mathcal{L}(v_{ini})$ in \mathcal{B} , denn Knotenbeschriftungen werden durch das Anwenden der Tableauregeln nicht gelöscht. Also ist wegen der obigen Behauptung $v_{ini} \in C_0^{\mathcal{I}}$, und damit ist C_0 erfüllbar.

Beweis der Behauptung per Induktion über die Struktur von C .

Induktionsanfang. Für Konzeptnamen $C = A$ gilt die Behauptung wegen der Definition von \mathcal{I} .

Induktionsschritt. Hier müssen wir für jeden der Operatoren $\neg, \sqcap, \sqcup, \exists, \forall$ einen gesonderten Fall betrachten, denn wegen der NNF kann man nun nicht mehr annehmen, dass \sqcup durch \sqcap oder \forall durch \exists ausgedrückt wird.

$C = \neg A$ für einen Konzeptnamen A . Da der I-Baum \mathcal{B} keinen offensichtlichen Widerspruch enthält, folgt aus $\neg A \in \mathcal{L}(v)$, dass $A \notin \mathcal{L}(v)$. Nach Definition von \mathcal{I} gilt dann aber $v \notin A^{\mathcal{I}}$, also $v \in (\neg A)^{\mathcal{I}}$.

$C = D_1 \sqcap D_2$. Wenn $C \in \mathcal{L}(v)$ in \mathcal{B} , dann auch $D_1 \in \mathcal{L}(v)$ und $D_2 \in \mathcal{L}(v)$, da \mathcal{B} vollständig ist und somit insbesondere die \sqcap -Regel nicht anwendbar ist. Mit der Induktionsvoraussetzung folgt daraus nun $v \in D_1^{\mathcal{I}}$ und $v \in D_2^{\mathcal{I}}$; wegen der Semantik von \sqcap gilt damit $v \in (D_1 \sqcap D_2)^{\mathcal{I}}$.

$C = D_1 \sqcup D_2$. Analog.

$C = \exists r.D$. Wegen $C \in \mathcal{L}(v)$ und da die \exists -Regel nicht anwendbar ist, gibt es $v' \in V$ mit $(v, r, v') \in E$ und $D \in \mathcal{L}(v')$. Nach Induktionsvoraussetzung folgt aus dem letzteren, dass $v' \in D^{\mathcal{I}}$. Da wegen Definition von \mathcal{I} auch $(v, v') \in r^{\mathcal{I}}$ sein muss, folgt $v \in (\exists r.D)^{\mathcal{I}}$ mit der Semantik von \exists .

$C = \forall r.D$. Ähnlich. \square

T4.9 Beispiel für Realisierbarkeit

Das folgende Bild zeigt einen I-Baum \mathcal{B} (links) und eine Interpretation \mathcal{I} , die \mathcal{B} realisiert, (rechts) sowie die zugehörige Abbildung π gemäß Definition 4.9 (gestrichelte Pfeile).



Man beachte, dass die Knoten von \mathcal{B} (links) mit Mengen von Konzepten beschriftet sind, aber die Elemente von \mathcal{I} (rechts) mit Konzeptnamen – so, wie wir Interpretationen immer darstellen.

T4.10 Verbleibender Beweis der Vollständigkeit

Behauptung. Für alle $i \leq n$ ist M_i realisierbar.

Beweis. Per Induktion über i .

Induktionsanfang. $M_0 = \{\mathcal{B}_{\text{ini}}\}$ ist realisierbar, da \mathcal{B}_{ini} realisierbar ist, und zwar durch ein beliebiges Modell von C_0 (welches existiert, da C_0 nach Voraussetzung von Theorem 4.10 erfüllbar ist).

Induktionsschritt. Wir nehmen an, dass M_i realisierbar sei, und zeigen, dass dann auch M_{i+1} realisierbar ist. Dazu unterscheiden wir vier Fälle gemäß der Regel, mit der M_{i+1} aus M_i erzeugt wird. Sei also $\mathcal{B} = (V, E, \mathcal{L})$ ein realisierbarer Baum aus M_i , auf den eine Regel angewendet wird.

\sqcup -Regel. Dann entsteht M_{i+1} aus M_i , indem \mathcal{B} durch zwei Bäume $\mathcal{B}' = (V, E, \mathcal{L}')$ und $\mathcal{B}'' = (V, E, \mathcal{L}'')$ ersetzt wird, und es gibt einen Knoten $v \in V$ mit

- $C \sqcup D \in \mathcal{L}(v)$;
- $\mathcal{L}'(v) = \mathcal{L}(v) \cup \{C\}$, $\mathcal{L}''(v) = \mathcal{L}(v) \cup \{D\}$;
- $\mathcal{L}'(u) = \mathcal{L}''(u) = \mathcal{L}(u)$ für alle $u \neq v$.

Es genügt zu zeigen: wenn \mathcal{B} realisierbar, dann \mathcal{B}' realisierbar *oder* \mathcal{B}'' realisierbar.

Sei \mathcal{I} eine Interpretation, die \mathcal{B} realisiert, und $\pi : V \rightarrow \Delta^{\mathcal{I}}$ die zugehörige Abbildung gemäß Definition 4.9. Dann gilt insbesondere $\pi(v) \in (C \sqcup D)^{\mathcal{I}}$. Nach Semantik von \sqcup ist dann $\pi(v) \in C^{\mathcal{I}}$ oder $\pi(v) \in D^{\mathcal{I}}$. Im ersten Fall realisiert \mathcal{I} den Baum \mathcal{B}' , im zweiten \mathcal{B}'' .

\sqcap -Regel. Analog, aber mit nur *einem* Baum \mathcal{B}' (probiert es selbst aus).

\exists -Regel. Dann entsteht M_{i+1} aus M_i , indem \mathcal{B} durch einen Baum $\mathcal{B}' = (V', E', \mathcal{L}')$ ersetzt wird, und es gibt Knoten $v, v' \in V$ mit

- $\exists r.C \in \mathcal{L}(v)$;
- $V' = V \uplus \{v'\}$, $E' = E \cup \{(v, r, v')\}$
- $\mathcal{L}'(v') = \{C\}$;
- $\mathcal{L}'(u) = \mathcal{L}(u)$ für alle $u \neq v'$.

Es genügt zu zeigen: wenn \mathcal{B} realisierbar, dann \mathcal{B}' realisierbar.

Sei \mathcal{I} eine Interpretation, die \mathcal{B} realisiert, und $\pi : V \rightarrow \Delta^{\mathcal{I}}$ die zugehörige Abbildung gemäß Definition 4.9. Dann gilt insbesondere $\pi(v) \in (\exists r.C)^{\mathcal{I}}$, also gibt es ein $d \in \Delta^{\mathcal{I}}$ mit $(\pi(v), d) \in r^{\mathcal{I}}$ und $d \in C^{\mathcal{I}}$. Wenn man nun π erweitert um $\pi(v') = d$, dann bezeugt die resultierende Funktion, dass \mathcal{B}' durch \mathcal{I} realisiert wird.

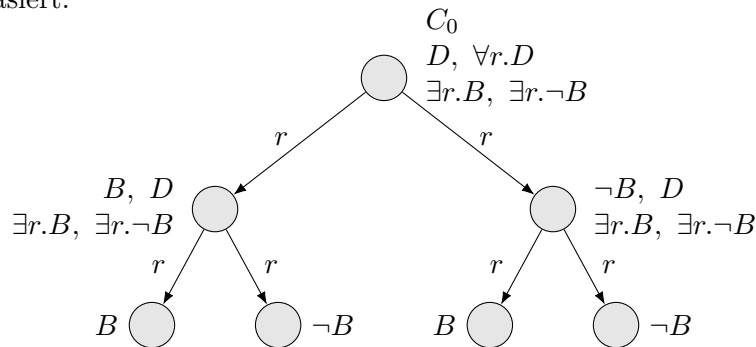
\forall -Regel. Ähnlich (probiert es selbst aus). □

T4.11 Beispiel für den Worst Case bei der Komplexitätsanalyse

Sei $n = 2$. Dann ist $C_0 = \forall r^0.(\exists r.B \sqcap \exists r.\neg B) \sqcap \forall r^1.(\exists r.B \sqcap \exists r.\neg B)$. (Folie 4.25). Kürzt man nun das doppelt vorkommende Teilkonzept ab mit $D = \exists r.B \sqcap \exists r.\neg B$, dann ist

$$C_0 = D \sqcap \forall r.D.$$

Der Tableau-Algorithmus erzeugt folgenden I-Baum, der auf dem vollständigen Binärbaum der Tiefe 2 basiert:



Es sollte nun leicht zu sehen sein, dass für beliebiges n der entsprechende I-Baum auf dem vollständigen Binärbaum der Tiefe n basiert, welcher 2^n Knoten hat.

T4.12 Beispiel für Backjumping

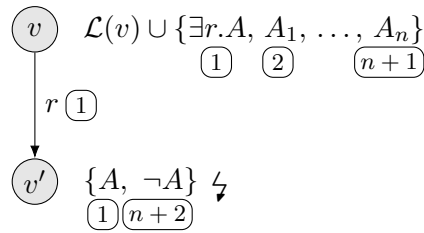
Nehmen wir an, ein Knoten v in einem I-Baum sei markiert wie folgt, wobei die Konzepte in $\mathcal{L}(v)$ durchnummeriert werden.

$$\mathcal{L}(v) = \{ \underbrace{(\exists r.A) \sqcup B}_{(1)}, \underbrace{A_1 \sqcup A'_1}_{(2)}, \dots, \underbrace{A_n \sqcup A'_n}_{(n+1)}, \underbrace{\forall r.\neg A}_{(n+2)} \}$$

Nehmen wir weiterhin an, der Tableau-Algorithmus verarbeite die Konzepte aus $\mathcal{L}(v)$ in der angegebenen Reihenfolge und wählt bei den Disjunktionen immer zuerst das linke Disjunkt. Dann wird

- im 1. Schritt das Konzept $\exists r.A$ zu $\mathcal{L}(v)$ hinzugefügt (\sqcup -Regel auf (1) angewendet);
- im 2. Schritt ein r -Nachfolger v' zu v hinzugefügt und mit $\{A\}$ markiert (\exists -Regel auf $\exists r.A \in \mathcal{L}(v)$ angewendet);
- im den nächsten n Schritten die linken Disjunkte von $(2) - (n+1)$ zu $\mathcal{L}(v)$ hinzugefügt (\sqcup -Regel);
- im folgenden Schritt das Konzept $\neg A$ zu $\mathcal{L}(v')$ hinzugefügt (\forall -Regel auf $\forall r.\neg A \in \mathcal{L}(v)$ angewendet).

Danach sieht die „Umgebung“ von v im I-Baum wie folgt aus, wobei die Einträge der Knotenlabels mit der Nummer des Konzeptes aus $\mathcal{L}(v)$ versehen sind, von dem sie stammen (dies repräsentiert die „Dependenzmenge“).



Wegen des offensichtlichen Widerspruchs in v' (angezeigt durch ζ) muss Backtracking durchgeführt werden. Einfaches Backtracking probiert hier aber 2^n sinnlose Alternativen aus, nämlich sämtliche Kombination der jeweils zwei Disjunkte von $(2) - (n+1)$, und findet erst dann die Ursache des Problems: Der Konflikt wurde letztlich durch das linke Disjunkt $\exists r.A$ von (1) und Konzept $\forall r.\neg A \ (n+2)$ ausgelöst. Mit Backjumping hingegen wird wie im Bild angedeutet Buch über die Herkunft der Konzepte geführt; deshalb kann direkt zur Ursache des Problems zurückgesprungen werden – hier zum 1. Schritt der obigen Aufzählung.

T4.13 Beispiel für Normalform der TBoxen

Wir betrachten die TBox

$$\mathcal{T} = \{A \sqsubseteq \exists r.B, A \sqcup B \sqsubseteq \forall r.B\}.$$

Beim Umwandeln in Normalform werden zuerst die beiden Konzeptinklusionen in eine einzige Konzeptinklusion mit \top auf der linken Seite umgewandelt, wie auf der Folie angegeben:

$$\{\top \sqsubseteq (\neg A \sqcup \exists r.B) \sqcap (\neg(A \sqcup B) \sqcup \forall r.B)\}$$

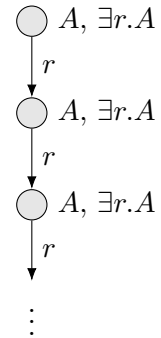
Nun muss noch NNF hergestellt werden, indem die rechte Negation „nach innen gezogen“ wird:

$$\mathcal{T}' = \{\top \sqsubseteq (\neg A \sqcup \exists r.B) \sqcap ((\neg A \sqcap \neg B) \sqcup \forall r.B)\}$$

Es ist leicht zu zeigen, dass die bei resultierende TBox \mathcal{T}' immer äquivalent zur ursprünglichen TBox \mathcal{T} ist, d. h. \mathcal{T} und \mathcal{T}' haben dieselben Modelle.

T4.14 Beispiel für Nicht-Terminierung

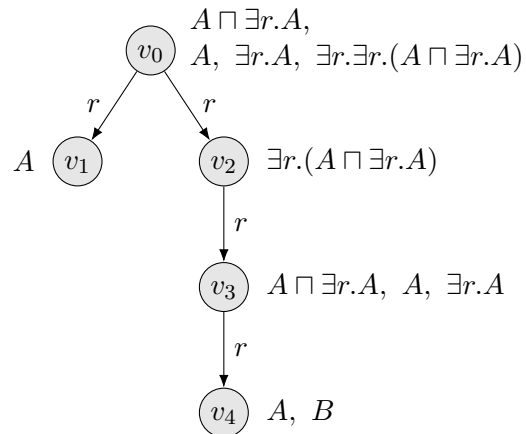
Sei $C_0 = A$ und $\mathcal{T} = \{\top \sqsubseteq \exists r.A\}$ (bereits in Normalform). Wenn man das bisherige Tableauverfahren um die TBox-Regel erweitert, terminiert der Algorithmus auf der Eingabe (C_0, \mathcal{T}) nicht mehr, denn es sind immer abwechselnd die TBox-Regel und die \exists -Regel anwendbar, wodurch der nebenstehende unendliche I-Baum erzeugt wird.



T4.15 Beispiel für Blockierung

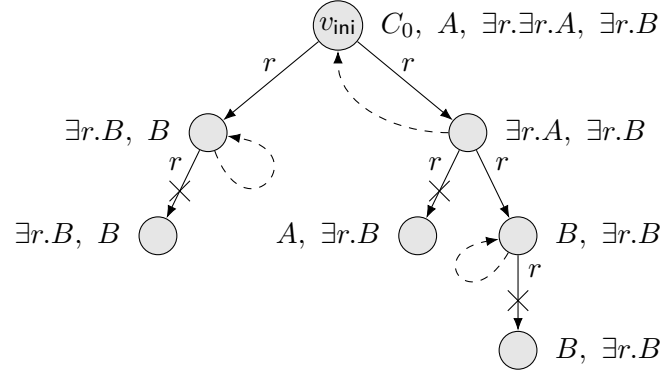
Im nebenstehenden I-Baum ist/sind

- v_1 direkt blockiert durch v_0 ;
- v_3 direkt blockiert durch v_0 ;
- v_1, v_3, v_4 blockiert;
- v_0, v_2 *nicht* blockiert.



T4.16 Beispiel für die \exists' -Regel

Seien $C_0 = A \sqcap \exists r. \exists r. A$ und $\mathcal{T} = \{\top \sqsubseteq \exists r. B\}$. Dann erzeugt das Tableauverfahren mit der neuen \exists' -Regel (und Blockierung) folgenden I-Baum.



Blockierte Knoten sind dadurch gekennzeichnet, dass die eingehende r -Kante durchkreuzt ist. Alle blockierten Knoten in diesem Beispiel sind direkt blockiert.

Die gestrichelten Pfeile zeigen an, durch welches Element der blockierte Nachfolger blockiert wird. Sie sind genau genommen nicht Bestandteil des I-Baums, aber sie werden bei der Modellkonstruktion im Korrektheitsbeweis nützlich sein.

T4.17 Verbleibender Beweis der Korrektheit (mit TBoxen)

Behauptung. Für alle Konzepte C und Knoten $v \in \Delta^{\mathcal{I}}$ gilt: $C \in \mathcal{L}(v)$ impliziert $v \in C^{\mathcal{I}}$

Man beachte, dass die Behauptung nur über die $v \in \Delta^{\mathcal{I}}$ eine Aussage macht, also über die im vollständigen I-Baum \mathcal{B} *unblockierten* Knoten.

Zunächst zeigen wir, dass diese Behauptung die Aussage des Theorems 4.13 impliziert.

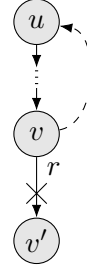
Da $C_0 \in \mathcal{L}(v_{\text{ini}})$, gilt nach der obigen Behauptung $v_{\text{ini}} \in C_0^{\mathcal{I}}$. Da außerdem die TBox-Regel nicht mehr anwendbar ist (\mathcal{B} ist vollständig), gilt $C_{\mathcal{T}} \in \mathcal{L}(v)$ für alle $v \in V$. Wegen der Behauptung ist somit \mathcal{I} auch ein Modell von \mathcal{T} .

Beweis der Behauptung per Induktion über die Struktur von C . Der Induktionsanfang ($C = A$) und die Fälle $C = \neg A$, $C = D_1 \sqcap D_2$, $C = D_1 \sqcup D_2$ im Induktionsschritt sind genauso wie in der Variante ohne TBoxen (T4.8). Für die übrigen zwei Fälle argumentieren wie folgt:

$C = \exists r. D$. Wegen $C \in \mathcal{L}(v)$ und da die \exists' -Regel nicht anwendbar ist, gibt es $v' \in V$ mit $(v, r, v') \in E$ und $D \in \mathcal{L}(v')$.

Falls v' unblockiert ist, folgt $(v, v') \in r^{\mathcal{I}}$ nach Definition von \mathcal{I} und $v' \in D^{\mathcal{I}}$ nach Induktionsvoraussetzung.

Falls v' blockiert ist, argumentieren wir wie folgt: da der direkte Vorgänger von v' der unblockierte Knoten v ist, gibt es einen *unblockierten* Vorgänger u von v' , durch den v' *direkt* blockiert ist, siehe nebenstehende Skizze. (Der Knoten u kann, aber muss nicht mit v übereinstimmen; siehe das vorangehende Beispiel). Nach Definition von \mathcal{I} ist $(v, u) \in r^{\mathcal{I}}$. Da außerdem $D \in \mathcal{L}(v') \subseteq \mathcal{L}(u)$ gilt, ist $u \in D^{\mathcal{I}}$ nach Induktionsvoraussetzung.



In beiden Fällen folgt $v \in (\exists r.D)^{\mathcal{I}}$.

C = $\forall r.D$. Um zu zeigen, dass $v \in (\forall r.D)^{\mathcal{I}}$, betrachten wir ein beliebiges Element u mit $(v, u) \in r^{\mathcal{I}}$ und müssen zeigen, dass $u \in D^{\mathcal{I}}$.

Nach Definition von $r^{\mathcal{I}}$ gibt es wegen $(v, u) \in r^{\mathcal{I}}$ zwei mögliche Fälle:

- (1) $(v, r, u) \in E$ für einen *unblockierten* Knoten u . Da die \forall -Regel nicht anwendbar ist, gilt $D \in \mathcal{L}(u)$, also nach Induktionsvoraussetzung $u \in D^{\mathcal{I}}$.
- (2) $(v, r, v') \in E$ für einen Knoten v' , der durch u direkt blockiert wird; siehe wieder die obige Skizze. Da die \forall -Regel nicht anwendbar ist, gilt $D \in \mathcal{L}(v')$. Da außerdem $\mathcal{L}(v') \subseteq \mathcal{L}(u)$, folgt auch hier $D \in \mathcal{L}(u)$ und nach Induktionsvoraussetzung $u \in D^{\mathcal{I}}$. □

T4.18 Tiefe der I-Bäume (mit TBoxen)

Behauptung. Es werden nur I-Bäume mit einer Tiefe von maximal 2^k generiert, wobei $k = |C_0| + |\mathcal{T}|$.

Beweis. Angenommen, es werde ein I-Baum der Tiefe $> 2^k$ erzeugt. Dazu muss irgendwann die \exists' -Regel auf einen Knoten v der Tiefe 2^k angewendet werden. Es gibt also einen Pfad v_0, \dots, v_{2^k} von der Wurzel $v_0 = v_{\text{ini}}$ zum Knoten $v_{2^k} = v$ (zeichnet den Baum und den Pfad selbst auf). Auf diesem Pfad gibt es $2^k + 1$ Knoten. Da es aber nur $2^{\text{sub}(C_0, \mathcal{T})} \leq 2^k$ mögliche Knotenbeschriftungen gibt, muss es nach dem Schubfachprinzip zwei verschiedene Knoten v_i, v_j auf dem Pfad geben mit derselben Beschriftung $\mathcal{L}(v_i) = \mathcal{L}(v_j)$. Also ist v_j blockiert und damit auch v , was ein Widerspruch zur oben gefolgerten Anwendung der \exists' -Regel ist. Damit ist die Annahme widerlegt und die Behauptung bewiesen. □

... Dokument wird fortgesetzt ...

Teil 5

Komplexität

Teil 6

Effiziente Beschreibungslogiken

Teil 7

ABoxen und Anfragebeantwortung

Literaturverzeichnis

- [BHLS17] Baader, Franz, Ian Horrocks, Carsten Lutz und Uli Sattler: *An Introduction to Description Logic*. Cambridge University Press, 2017. <https://tinyurl.com/suub-intro-dl-ebook>.