

Vorlesungsübersicht

Beschreibungslogik

Kapitel 6: Effiziente Beschreibungslogiken

Sommersemester 2017 Thomas Schneider

AG Theorie der künstlichen Intelligenz (TdKI)

<http://tinyurl.com/ss17-b1>

Kapitel 1: Einleitung

Kapitel 2: Grundlagen

Kapitel 3: Ausdrucksstärke und Modellkonstruktionen

Kapitel 4: Tableau-Algorithmen

Kapitel 5: Komplexität

Kapitel 6: Effiziente Beschreibungslogiken

Kapitel 7: ABoxen und Anfragebeantwortung

Ziel des Kapitels

Für manche Anwendungen ist \mathcal{ALC} zu komplex:

- Auch hoch-optimierte Reasoner können sehr große und komplexe Ontologien oft nicht verarbeiten (oder nur nach intensivem Tuning)
- In der Anfragebeantwortung muss man oft mit sehr großen Datenmengen umgehen und braucht schnelle Antworten (Kapitel 7)

Wir betrachten die Beschreibungslogik \mathcal{EL} :

- viel weniger ausdrucksstark als \mathcal{ALC} , Basisoperatoren nur \sqcap und $\exists r.C$
- Erfüllbarkeit und Subsumtion in Polyzeit entscheidbar

Kapitel 6: Effiziente Beschreibungslogiken

① \mathcal{EL}

② Subsumtion ohne TBox

③ Subsumtion mit TBoxen

④ Erweiterungen von \mathcal{EL}

Kapitel 6: Effiziente Beschreibungslogiken

- 1 EL
- 2 Subsumtion ohne TBox
- 3 Subsumtion mit TBoxen
- 4 Erweiterungen von EL

Simulation

Intuitiv: EL ist die „Hälfte von ALC“;
Simulation entspricht der „Hälfte von Bisimulation“.

Definition 6.2 (Simulation)

Seien \mathcal{I}_1 und \mathcal{I}_2 Interpretationen.

Relation $\rho \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ ist **Simulation** von \mathcal{I}_1 nach \mathcal{I}_2 , wenn gilt:

- 1 Wenn $d_1 \rho d_2$, dann gilt für alle Konzeptnamen A :

$$d_1 \in A^{\mathcal{I}_1} \quad \text{impliziert} \quad d_2 \in A^{\mathcal{I}_2}$$

- 2 Wenn $d_1 \rho d_2$ und $(d_1, d'_1) \in r^{\mathcal{I}_1}$ für beliebigen Rollennamen r , dann gibt es ein $d'_2 \in \Delta^{\mathcal{I}_2}$ mit $d'_1 \rho d'_2$ und $(d_2, d'_2) \in r^{\mathcal{I}_2}$.

T 6.1

Beachte:

Im Gegensatz zu Bisimulationen sind **Simulationen gerichtet**.

EL

Definition 6.1 (EL)

Ein **EL-Konzept** ist ein **ALC-Konzept**, in dem nur die Konstruktoren \top , \sqcap und $\exists r.C$ verwendet werden.

EL ist beliebt für biomedizinische Ontologien
(die sind oft groß und mit hohem Abstraktionsgrad):

Perikardium \sqsubseteq Gewebe \sqcap \exists teilVon.Herz
 Perikarditis \equiv Entzündung \sqcap \exists ort.Perikardium
 Entzündung \sqsubseteq Krankheit \sqcap \exists wirktAuf.Gewebe

SNOMED CT ist in unwesentlicher Erweiterung von EL formuliert.

EL ist Grundlage des **EL-Profiles von OWL 2**.

Simulation

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

Wir schreiben $(\mathcal{I}_1, d_1) \lesssim (\mathcal{I}_2, d_2)$,

wenn es Simulation ρ von \mathcal{I}_1 nach \mathcal{I}_2 gibt mit $d_1 \rho d_2$
(wir sagen: d_1 wird simuliert von d_2).

Theorem 6.3

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

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

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

T 6.2

Simulation vs. Bisimulation; Ausdruckstärke

Intuitiv: Wenn $(\mathcal{I}_1, d_1) \lesssim (\mathcal{I}_2, d_2)$ und $(\mathcal{I}_2, d_2) \lesssim (\mathcal{I}_1, d_1)$, dann kann \mathcal{EL} nicht zwischen d_1 und d_2 „unterscheiden“.

Achtung:

Bisimulation und wechselseitige Simulation sind **nicht dasselbe**:

Lemma 6.4

Es gibt (\mathcal{I}_1, d_1) und (\mathcal{I}, d_2) , so dass:

- $(\mathcal{I}_1, d_1) \lesssim (\mathcal{I}_2, d_2)$ und $(\mathcal{I}_2, d_2) \lesssim (\mathcal{I}_1, d_1)$
- $(\mathcal{I}_1, d_1) \not\sim (\mathcal{I}_2, d_2)$

T 6.3

Man kann nun wieder Nicht-Ausdrückbarkeitsresultate zeigen:

Lemma 6.5

Das \mathcal{ALC} -Konzept $\forall r.A$ ist **nicht** in \mathcal{EL} ausdrückbar.

T 6.3 Forts.

Kapitel 6: Effiziente Beschreibungslogiken

- 1 \mathcal{EL}
- 2 Subsumtion ohne TBox
- 3 Subsumtion mit TBoxen
- 4 Erweiterungen von \mathcal{EL}

Schlussfolgerungsprobleme in \mathcal{EL}

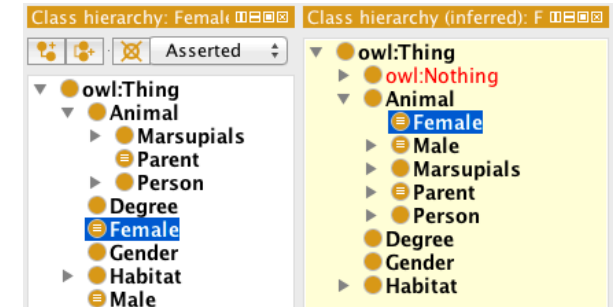
In \mathcal{EL} ist Erfüllbarkeit **kein** interessantes Schlussfolgerungsproblem:

Lemma 6.6

Jedes \mathcal{EL} -Konzept ist erfüllbar bzgl. jeder TBox.

T 6.4

Darum konzentrieren wir uns auf **Subsumtion**.

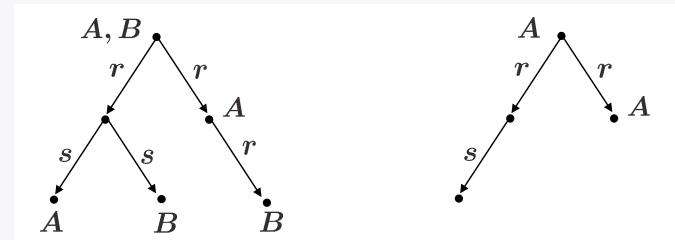


Intuition

Eine Subsumtion $C \sqsubseteq D$ gilt in \mathcal{EL} im Prinzip genau dann, wenn man **D syntaktisch „in C wiederfindet“**.

$$\begin{array}{ll}
 \text{z. B.: } & C = A \sqcap B & D = A \\
 & \sqcap \exists r. (\exists s. A \sqcap \exists s. B) & \sqcap \exists r. \exists s. T \\
 & \sqcap \exists r. (A \sqcap \exists r. B) & \sqcap \exists r. A
 \end{array}$$

Konzepte dargestellt als Bäume: „Wiederfinden“ entspricht **Simulation** von D-Baum in C-Baum (**Richtung!**)

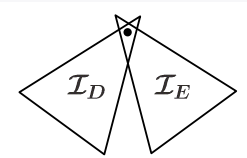


Kanonisches Modell

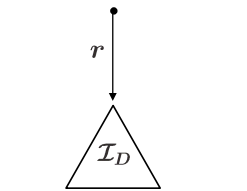
Definition 6.7

Wir ordnen induktiv jedem \mathcal{EL} -Konzept C eine Interpretation \mathcal{I}_C zu:

$C = \top$ Modell \mathcal{I}_C : •
 $C = A$ Modell \mathcal{I}_C : • A



$C = D \sqcap E$ Modell \mathcal{I}_C :



$C = \exists r.D$ Modell \mathcal{I}_C :

Kanonisches Modell

Wir nennen \mathcal{I}_C **kanonisches Modell**, bezeichnen Wurzel stets mit d_W .

Man sieht leicht, dass \mathcal{I}_C wirklich ein **Modell** ist:

Lemma 6.8

Für alle \mathcal{EL} -Konzepte C gilt:

Die Interpretation \mathcal{I}_C ist Modell von C mit $d_W \in C^{\mathcal{I}_C}$.

T 6.5

Die zentrale Eigenschaft kanonischer Modelle:

Lemma 6.9

Für alle \mathcal{EL} -Konzepte C , Interpretationen \mathcal{I} und $e \in \Delta^{\mathcal{I}}$ gilt:

$$e \in C^{\mathcal{I}} \text{ gdw. } (\mathcal{I}_C, d_W) \lesssim (\mathcal{I}, e)$$

T 6.6

Charakterisierung von Subsumtion

Wir zeigen nun, dass $C \sqsubseteq D$ gdw. man \mathcal{I}_D in \mathcal{I}_C "wiederfindet":

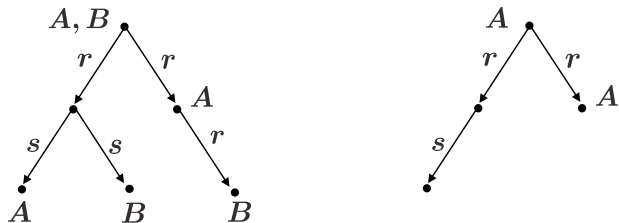
Lemma 6.10

Für alle \mathcal{EL} -Konzepte C, D gilt:

$$C \sqsubseteq D \text{ gdw. } (\mathcal{I}_D, d_W) \lesssim (\mathcal{I}_C, d_W)$$

T 6.7

z. B.: $C = A \sqcap B$ $D = A$
 $\sqcap \exists r. (\exists s. A \sqcap \exists s. B)$ $\sqcap \exists r. \exists s. \top$
 $\sqcap \exists r. (A \sqcap \exists r. B)$ $\sqcap \exists r. A$



Entscheidungsverfahren für Subsumtion

Folgendes Theorem formuliert den (recht einfachen) Algorithmus:

Theorem 6.11

Subsumtion in \mathcal{EL} kann in polynomieller Zeit entschieden werden:

- Konstruiere \mathcal{I}_C und \mathcal{I}_D in polynomieller Zeit.
- Überprüfe in polynomieller Zeit, ob $(\mathcal{I}_D, d_W) \lesssim (\mathcal{I}_C, d_W)$.

T 6.8

Kapitel 6: Effiziente Beschreibungslogiken

- 1 εL
- 2 Subsumtion ohne TBox
- 3 Subsumtion mit TBoxen
- 4 Erweiterungen von εL

Vereinfachende Annahme 1

Betrachten **o. B. d. A.** nur Subsumtion von Konzeptnamen bzgl. TBoxen.

Lemma 6.12

Seien C, D zwei beliebige εL-Konzepte und \mathcal{T} eine εL-TBox.

Sei weiterhin $\mathcal{T}' = \mathcal{T} \cup \{A_C \sqsubseteq C, D \sqsubseteq A_D\}$,
mit Konzeptnamen A_C, A_D , die nicht in C, D, \mathcal{T} vorkommen.

Dann gilt:

$$\mathcal{T} \models C \sqsubseteq D \quad \text{gdw.} \quad \mathcal{T}' \models A_C \sqsubseteq A_D$$

(Übung)

Lemma 6.12 liefert **Polyzeit-Reduktion**

- von Subsumtion zwischen beliebigen Konzepten bzgl. TBoxen
- auf Subsumtion zwischen Konzeptnamen bzgl. TBoxen.

Intuition

Wir verwenden ein so genanntes **konsequenzbasiertes** Verfahren.

Grundidee:

- Mit Hilfe von **Regeln** werden zur TBox nach und nach neue Konzeptinklusionen hinzugefügt.
- Am Ende muss man dann nur noch nachschauen, ob die gewünschte Subsumtion in der TBox **explizit enthalten** ist.

In der Praxis haben sich derartige Verfahren als **äußerst effizient** herausgestellt.

Sie sind verwandt mit **Sequenzkalkülen** aus der klassischen Logik.

Annahme 2: Normalform

Betrachten **o. B. d. A.** nur TBoxen in folgender Normalform:

Definition 6.13

Eine TBox ist in **Normalform (NF)**, wenn sie nur Inklusionen der Form

$$A_1 \sqcap \dots \sqcap A_n \sqsubseteq A \quad A \sqsubseteq \exists r.A_1 \quad \exists r.A \sqsubseteq A_1$$

enthält, wobei A, A_1, \dots, A_n Konzeptnamen oder \top sind.

Lemma 6.14

Jede εL-TBox \mathcal{T} kann in polynomieller Zeit in eine TBox \mathcal{T}' in NF gewandelt werden, so dass für alle Konzeptnamen A, B in \mathcal{T} gilt:

$$\mathcal{T} \models A \sqsubseteq B \quad \text{gdw.} \quad \mathcal{T}' \models A \sqsubseteq B \quad (*)$$

Wenn (*) gilt, sagen wir: \mathcal{T}' ist **konservative Erweiterung** von \mathcal{T} .

Um die NF herzustellen, wenden wir **Normalisierungsregeln** an. **T 6.9**

Normalisierung

NF1 $C_1 \sqcap \dots \sqcap C_n \sqsubseteq E \rightsquigarrow C_i \sqsubseteq A_{C_i}, C_1 \sqcap \dots \sqcap C_{i-1} \sqcap A_{C_i} \sqcap C_{i+1} \sqcap \dots \sqcap C_n \sqsubseteq E$
wenn C_i Existenzrestriktion

NF2 $\exists r.C \sqsubseteq E \rightsquigarrow C \sqsubseteq A_C, \exists r.A_C \sqsubseteq E$

NF3 $C \sqsubseteq \exists r.D \rightsquigarrow C \sqsubseteq A_C, A_C \sqsubseteq \exists r.D$

NF4 $A \sqsubseteq \exists r.C \rightsquigarrow A \sqsubseteq \exists r.A_C, A_C \sqsubseteq C$

NF5 $A \sqsubseteq C_1 \sqcap C_2 \rightsquigarrow A \sqsubseteq C_1, A \sqsubseteq C_2$

} wenn C weder Konzeptname noch T

A ist Konzeptname; C_i, D, E sind beliebige Konzepte; A_{C_i} sind neue Konzeptnamen.

Lemma 6.15
Jede εL-TBox \mathcal{T} kann durch **linear viele** Regelanwendungen in TBox in NF transformiert werden, die konservative Erweiterung von \mathcal{T} ist.

T 6.10

Saturierung

Für eine εL-TBox \mathcal{T} sei \mathcal{T}^* das Ergebnis erschöpfender Regelanwendung. Wir nennen \mathcal{T}^* die **Saturierung** von \mathcal{T} .

\mathcal{T}^* macht alle Subsumtionen zwischen Konzeptnamen explizit:

Theorem 6.16
Für alle Konzeptnamen A, B in \mathcal{T} gilt:
$$\mathcal{T} \models A \sqsubseteq B \text{ gdw. } A \sqsubseteq B \in \mathcal{T}^*$$

Der Algorithmus **klassifiziert** also die Konzeptnamen vollständig; berechnet nicht nur eine einzelne Subsumtion.

Theorem 6.17
Die Konstruktion von \mathcal{T}^* **terminiert** nach $\mathcal{O}(|\mathcal{T}|^2)$ vielen Regelanwendungen.

T 6.12

Subsumtion mit TBox

Der Algorithmus beginnt mit der ursprünglichen TBox und wendet dann erschöpfend folgende Regeln an.

R1 $\frac{}{A \sqsubseteq A}$ wenn A in \mathcal{T} vorkommt

R2 $\frac{}{A \sqsubseteq T}$ wenn A in \mathcal{T} vorkommt

R3 $\frac{A \sqsubseteq A_1 \quad \dots \quad A \sqsubseteq A_n \quad A_1 \sqcap \dots \sqcap A_n \sqsubseteq B}{A \sqsubseteq B}$

R4 $\frac{A \sqsubseteq \exists r.A_1 \quad A_1 \sqsubseteq B_1 \quad \exists r.B_1 \sqsubseteq B}{A \sqsubseteq B}$

T 6.11

Beachte: A, A_1, \dots, A_n, B sind Konzeptnamen oder T.

Korrektheit des Algorithmus

Wir beweisen nun Theorem 6.16. Korrektheit:

Lemma 6.18
Für alle Konzeptnamen A, B in \mathcal{T} gilt:
$$A \sqsubseteq B \in \mathcal{T}^* \text{ impliziert } \mathcal{T} \models A \sqsubseteq B$$

Beweis.
Sei $\mathcal{T} = \mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_n = \mathcal{T}^*$ die durch Regelanwendung erzeugte Folge von TBoxen. Es genügt zu zeigen:

Beh.: Für alle $i < n$ gilt: $\mathcal{T}_i \models \mathcal{T}_{i+1}$
(d. h.: $\mathcal{T}_i \models C \sqsubseteq D$ für alle $C \sqsubseteq D \in \mathcal{T}_{i+1}$)

T 6.13

Daraus folgt direkt $\mathcal{T} \models \mathcal{T}^*$. □

Vollständigkeit

Konstruieren ein einziges Modell, das **alle nicht** aus \mathcal{T} folgenden Subsumtionen zwischen Konzeptnamen **gleichzeitig** falsch macht.

Existenz solcher **kanonischer Modelle** ist **zentrale Eigenschaft** von \mathcal{EL} .

Definition 6.19

Die *kanonische Interpretation* \mathcal{I} ist wie folgt definiert:

$$\begin{aligned} \Delta^{\mathcal{I}} &= \{d_A \mid A \text{ Konzeptname in } \mathcal{T}^*\} \cup \{d_{\top}\} \\ A^{\mathcal{I}} &= \{d_B \mid B \sqsubseteq A \in \mathcal{T}^*\} \\ r^{\mathcal{I}} &= \{(d_A, d_B) \mid A \sqsubseteq A' \in \mathcal{T}^* \text{ und } A' \sqsubseteq \exists r.B \in \mathcal{T}^*, \\ &\quad A' \text{ Konzeptname}\} \quad \mathbf{T6.14} \end{aligned}$$

Lemma 6.20

Die kanonische Interpretation \mathcal{I} ist ein Model von \mathcal{T}^* . **T6.15**

Kapitel 6: Effiziente Beschreibungslogiken

- 1 \mathcal{EL}
- 2 Subsumtion ohne TBox
- 3 Subsumtion mit TBoxen
- 4 Erweiterungen von \mathcal{EL}

Vollständigkeit

Jetzt zeigt man leicht:

Lemma 6.21

Für alle Konzeptnamen A, B in \mathcal{T} gilt:

$$\mathcal{T} \models A \sqsubseteq B \text{ impliziert } A \sqsubseteq B \in \mathcal{T}^*$$

Beweis. Angenommen $A \sqsubseteq B \notin \mathcal{T}^*$. (*)

Betrachte Element d_A der kanonischen Interpretation \mathcal{I} .

Wegen **R1** ist $A \sqsubseteq A \in \mathcal{T}^*$, also $d_A \in A^{\mathcal{I}}$.

Def. von \mathcal{I} und (*) liefert $d_A \notin B^{\mathcal{I}}$.

Da \mathcal{I} Modell von \mathcal{T}^* (Lemma 6.20) und damit von \mathcal{T} ist, folgt $\mathcal{T} \not\models A \sqsubseteq B$. □

„Gute“ und „schlechte“ Erweiterungen

👍 **Der Algorithmus kann angepasst werden an \mathcal{EL} erweitert mit:**

- \perp
- „Range Restrictions“ $\top \sqsubseteq \forall r.C$ und „Domain Restrictions“ $\top \sqsubseteq \forall r^-.C$
- Rolleninklusionen mit Verkettung: $r_1 \circ \dots \circ r_n \sqsubseteq r$
- ...

Dies (und mehr) ist im EL-Profil von OWL 2 realisiert.

👎 **Viele andere Erweiterungen sind jedoch EXPTIME-vollständig (wie \mathcal{ALC}).**

Wir betrachten exemplarisch

- \mathcal{ELU} , die Erweiterung von \mathcal{EL} mit \sqcup
- \mathcal{EL}_{\forall} , die Erweiterung von \mathcal{EL} mit $\forall r.C$
- $\mathcal{EL}^{\geq 2}$, die Erweiterung von \mathcal{EL} mit $\geq 2 r.T$

\mathcal{EL} mit Disjunktion und \perp

Theorem 6.22

Erfüllbarkeit in \mathcal{ELU}_\perp bzgl. TBoxen ist EXPTIME-vollständig.

Beweis.

Reduktion von **Erfüllbarkeit** von Konzeptnamen A bzgl. \mathcal{ALC} -TBox \mathcal{T}

Schritt 1. Ersetze in \mathcal{T} alle Werte- durch Existenzrestriktionen:

$$\forall r.C \quad \text{wird} \quad \neg \exists r. \neg C$$

Schritt 2. Modifiziere \mathcal{T} so, dass \neg nur vor Konzeptnamen auftritt:

$$\begin{aligned} \text{z. B. } A \sqsubseteq \exists s.(B' \sqcup \neg \exists r.B) \quad \text{wird} \quad A \sqsubseteq \exists s.(B' \sqcup \neg X) \\ X \equiv \exists r.B \\ (X \text{ neuer Konzeptname}) \end{aligned}$$

\mathcal{EL} mit Disjunktion

Theorem 6.24

Subsumtion in \mathcal{ELU} bzgl. TBoxen ist EXPTIME-vollständig.

Beweis.

Reduktion von **Erfüllbarkeit** von Konzeptnamen A bzgl. \mathcal{ELU}_\perp -TBox \mathcal{T}

Konstruiere \mathcal{ELU} -TBox \mathcal{T}' :

- nimm o. B. d. A. an, dass \perp nur in der Form $C \sqsubseteq \perp$ vorkommt (jedes \mathcal{ELU}_\perp -Konzept ist äquivalent zu \mathcal{ELU} -Konzept oder \perp)
- ersetze \perp durch neuen Konzeptnamen L T 6.17
- füge hinzu:

$$\exists r.L \sqsubseteq L \quad \text{für alle Rollennamen } r \text{ in } \mathcal{T}$$

Lemma 6.25

A unerfüllbar bzgl. \mathcal{T} **gdw.** $\mathcal{T}' \models A \sqsubseteq L$. T 6.18

\mathcal{EL} mit Disjunktion und \perp

Schritt 3. Entferne Negation vollständig aus \mathcal{T} :

- Ersetze jedes $\neg X$ durch neuen Konzeptnamen \bar{X}
- Erzwinge korrektes Verhalten der \bar{X} :

$$\begin{aligned} \top &\sqsubseteq X \sqcap \bar{X} \\ X \sqcup \bar{X} &\sqsubseteq \perp \end{aligned}$$

Die resultierende \mathcal{ELU}_\perp -TBox sei \mathcal{T}' .

Lemma 6.23

Für alle Konzeptnamen A gilt:

$$A \text{ erfüllbar bzgl. } \mathcal{T} \quad \text{gdw.} \quad A \text{ erfüllbar bzgl. } \mathcal{T}'$$

T 6.16

\mathcal{EL} mit Wertrestriktionen

\mathcal{EL}^\forall ist \mathcal{EL} erweitert um $\forall r.C$.

Theorem 6.26

In \mathcal{EL}^\forall ist Subsumtion bzgl. TBoxen EXPTIME-vollständig.

Beweis:

Reduktion von **Subsumtion** zwischen Konzeptnamen bzgl. \mathcal{ELU} -TBox \mathcal{T}

Können annehmen, dass \sqcup nur in den folgenden Formen vorkommt:

- $A_1 \sqcup A_2 \sqsubseteq A \rightsquigarrow$ Ersetze durch $A_1 \sqsubseteq A, A_2 \sqsubseteq A$
- $A \sqsubseteq B_1 \sqcup B_2 \rightsquigarrow$ Ersetze durch $A \sqcap \exists r.T \sqsubseteq B_1$
 $A \sqcap \forall r.X \sqsubseteq B_2$ r, X neu

Die resultierende \mathcal{EL}^\forall -TBox sei \mathcal{T}' .

Lemma 6.27

$\mathcal{T} \models A \sqsubseteq B$ **gdw.** $\mathcal{T}' \models A \sqsubseteq B$. T 6.19

EL mit Zahlenrestriktionen

$\mathcal{EL}^{\geq 2}$ ist EL erweitert um $\geq 2 r.T$.

Theorem 6.28

In $\mathcal{EL}^{\geq 2}$ ist Subsumtion bzgl. TBoxen EXP TIME-vollständig.

Beweis: Wieder Reduktion von **ELU-Subsumtion**

Können annehmen, dass \sqcup nur in den folgenden Formen vorkommt:

- $A_1 \sqcup A_2 \sqsubseteq A \rightsquigarrow$ Ersetze durch $A_1 \sqsubseteq A, A_2 \sqsubseteq A$
- $A \sqsubseteq B_1 \sqcup B_2 \rightsquigarrow$ Ersetze durch $A \sqsubseteq \exists r.X \sqcap \exists r.Y$
 $A \sqcap \exists r.(X \sqcap Y) \sqsubseteq B_1$
 $(r, X, Y \text{ neu}) \quad A \sqcap \geq 2 r.T \sqsubseteq B_2$

Die resultierende \mathcal{EL}^{\forall} -TBox sei \mathcal{T}' .

Lemma 6.29

$\mathcal{T} \models A \sqsubseteq B$ **gdw.** $\mathcal{T}' \models A \sqsubseteq B$. (Übung)

Zusammenfassung für EL

Die EL-Familie von BLen:

- Erlaubt Schlussfolgern in polynomieller Zeit
- Es gibt viele Reasoner wie ELK, CEL, SNOROCKET
- Skaliert auch auf große Terminologien wie SNOMED CT (> 400.000 Konzepte, wird in wenigen Sekunden klassifiziert)
- Stellt viele Operatoren zur Verfügung, hat aber eingeschränktes Ausdrucksvermögen (z. B. kann keine Disjunktion ausgedrückt werden – Konvexität!)

Konvexität

Eine Erweiterung von EL ist **konvex**, wenn für alle TBoxen \mathcal{T} und Konzepte C, D_1, D_2 gilt:

$\mathcal{T} \models C \sqsubseteq D_1 \sqcup D_2$ impliziert $\mathcal{T} \models C \sqsubseteq D_1$ oder $\mathcal{T} \models C \sqsubseteq D_2$

\mathcal{EL}^{\forall} ist **nicht konvex**:

$\mathcal{T} \models T \sqsubseteq \exists r.T \sqcup \forall r.X$ aber $\mathcal{T} \not\models T \sqsubseteq \exists r.T$
 und $\mathcal{T} \not\models T \sqsubseteq \forall r.X$

Unsere Beweise zeigen im Prinzip:

jede **nicht-konvexe** Erweiterung von EL ist **EXP TIME-hart**.

Aber auch konvexe Erweiterungen sind nicht zwangsläufig in PTIME:

Z. B. ist **ELI** (EL „plus“ $\exists r^- . C$) konvex, aber EXP TIME-vollst.

Für konvexe Erweiterungen gibt es oft effiziente konsequenzbasierte Algorithmen.

Literatur für dieses Kapitel (Basis)



Franz Baader, Ian Horrocks, Carsten Lutz, Uli Sattler.

An Introduction to Description Logic.

Cambridge University Press, 2017.

Kapitel 6: Reasoning in the EL Family of DLs

In SUUB verfügbar: <http://tinyurl.com/suub-intro-dl>

Bestellung beim Verlag: <http://www.cambridge.org/9780521695428>

Ich habe auch ein Exemplar – meldet Euch bei Bedarf gern.

Literatur für dieses Kapitel (weiterführend 1)



Franz Baader, Sebastian Brandt, Carsten Lutz.

Pushing the εℒ Envelope.

IJCAI 2005: 364–369.

<http://ijcai.org/Proceedings/05/Papers/0372.pdf>

Untersucht systematisch Erweiterungen von εℒ, die in Polyzeit bleiben oder höhere Komplexität verursachen.

Technischer Report mit Beweisdetails: <http://lat.inf.tu-dresden.de/research/reports/2005/BaaderBrandtLutz-LTCS-05-01.ps.gz>



Franz Baader, Sebastian Brandt, Carsten Lutz.

Pushing the εℒ Envelope Further.

OWLED (Spring) 2008.

http://ceur-ws.org/Vol-496/owlled2008dc_paper_3.pdf

Führt die vorangegangene Arbeit fort.

Links für dieses Kapitel



University of Manchester

OWL API

<http://owlcs.github.io/owlapi/>

<https://github.com/owlcs/owlapi/wiki> (Wiki + Doku.)

Die API der Wahl, um mit Ontologien zu arbeiten.



University of Manchester

List of Reasoners

<http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/>

Eine aktuelle Übersicht von DL-Reasonern

(die speziell für εℒ entwickelten haben meist ein „EL“ im Namen)

Literatur für dieses Kapitel (weiterführend 2)



Boontawee Suntisrivaraporn.

Optimization and Implementation of Subsumption Algorithms for the Description Logic εℒ with Cyclic TBoxes and General Concept Inclusion Axioms.

Masterarbeit, TU Dresden, 2005.

<https://lat.inf.tu-dresden.de/research/mas/Sun-Mas-05.pdf>

Erklärt anschaulich Normalisierung und Klassifikation für εℒ (ohne Erweiterungen). Allerdings weichen die Regeln etwas von denen aus der Vorlesung ab; „unsere“ sind besser optimiert.