Konjunktive Anfragen
und Beschreibungslogik-TBoxen

Ontology-Mediated Queries

Wir betrachten die Beantwortung konjunktiver Anfragen über *ABoxen* (statt Interpretationen in Gegenwart von TBoxen).

Dazu möchten wir idealerweise Datenbanksysteme verwenden.

Es macht für diesen Zweck Sinn,

- *nicht* TBox und ABox zu einer Wissensbasis zusammenzufassen,
- *sondern* CQ und TBox zu einer erweiterten Anfrage.

Dies führt zum Begriff einer *Ontology-Mediated Query.*

(auf deutsch also etwa: Ontologie-vermittelte Anfrage)

Ontology-Mediated Queries

Definition 7.9 (OMQ)
Ontology-mediated query (OMQ) ist Paar $\boldsymbol{Q} = (\boldsymbol{q}(\overline{\boldsymbol{x}}), \boldsymbol{\mathcal{T}})$
mit $\boldsymbol{q}(\overline{\boldsymbol{x}})$ konjunktiver Anfrage und $\boldsymbol{\mathcal{T}}$ TBox.

Antwort auf $\boldsymbol{Q}$ *in ABox* $\boldsymbol{\mathcal{A}}$: Tupel $\overline{\boldsymbol{a}} = \boldsymbol{a}_1 \cdots \boldsymbol{a}_n$ von Individuen,
das Antwort auf $\boldsymbol{q}(\overline{\boldsymbol{x}})$ in *allen* Modellen von $\boldsymbol{\mathcal{A}}$ und $\boldsymbol{\mathcal{T}}$ ist.

Menge aller Antworten auf $\boldsymbol{Q}$ in $\boldsymbol{\mathcal{A}}$ bezeichnen wir mit $\mathsf{cert}(\boldsymbol{Q}, \boldsymbol{\mathcal{A}})$.
Solche Antworten nennt man auch *sichere Antworten* (engl.: certain answers).

Beachte: $$\mathsf{cert}(\boldsymbol{Q}, \boldsymbol{\mathcal{A}}) = \bigcap_{\boldsymbol{\mathcal{I}} \text{ Modell von } \boldsymbol{\mathcal{A}} \text{ und } \boldsymbol{\mathcal{T}}} \mathsf{ans}(\boldsymbol{q}, \boldsymbol{\mathcal{I}})$$

T7.7

Beantwortung von OMQs

Anfragebeantwortung:

Gegeben $\boldsymbol{Q} = (\boldsymbol{q}, \boldsymbol{\mathcal{T}})$ und $\boldsymbol{\mathcal{A}}$, berechne cert$(\boldsymbol{Q}, \boldsymbol{\mathcal{A}})$.

Bei *Datenkomplexität* betrachten wir wieder nur die Daten (also $\boldsymbol{\mathcal{A}}$) als Eingabe, aber $\boldsymbol{Q}$ (und damit sowohl $\boldsymbol{q}$ als auch $\boldsymbol{\mathcal{T}}$) als fest.

Boolesche OMQ $\boldsymbol{Q} = (\boldsymbol{q}, \boldsymbol{\mathcal{T}})$ liefert für jede ABox $\boldsymbol{\mathcal{A}}$ entweder

- leeres Tupel () als einzige Antwort, notiert $\boldsymbol{\mathcal{A}} \models \boldsymbol{Q}$
  ($\boldsymbol{q} \rightarrow \boldsymbol{\mathcal{I}}$ für alle Modelle $\boldsymbol{\mathcal{I}}$ von $\boldsymbol{\mathcal{A}}$ und $\boldsymbol{\mathcal{T}}$) oder
- keine Antwort, notiert $\boldsymbol{\mathcal{A}} \not\models \boldsymbol{Q}$
  ($\boldsymbol{q} \not\rightarrow \boldsymbol{\mathcal{I}}$ für mindestens ein Modell $\boldsymbol{\mathcal{I}}$ von $\boldsymbol{\mathcal{A}}$ und $\boldsymbol{\mathcal{T}}$).

Für Boolesche OMQs ist Anfragebeantwortung also ein *Entscheidungsproblem.*

Datenkomplexität von OMQs

Wollen zeigen: Beantwortung konjunktiver Anfragen in $\mathcal{ALC}$ ist coNP-hart bzgl. Datenkomplexität.

Per Reduktion vom Komplement von 3-Färbbarkeit für ungerichtete Graphen.

Zur Erinnerung:
Ein ungerichteter Graph $G = (V, E)$ ist *3-färbbar*, wenn es eine Abbildung $f : V \to \{R, G, B\}$ gibt, so dass $f(v_1) \neq f(v_2)$ für alle $\{v_1, v_2\} \in E$.

Betrachte folgende OMQ $Q = (q, \mathcal{T})$ (TBox in $\mathcal{ALC}$):

$$\mathcal{T} = \{ \quad \top \sqsubseteq R \sqcup G \sqcup B, \quad R \sqcap \exists r.R \sqsubseteq D, \quad G \sqcap \exists r.G \sqsubseteq D, \quad B \sqcap \exists r.B \sqsubseteq D \quad \}$$

$$q = \exists x\, D(x)$$

D steht für „Defekt“

Datenkomplexität von OMQs

$$\mathcal{T} = \{ \quad \top \sqsubseteq R \sqcup G \sqcup B, \quad R \sqcap \exists r.R \sqsubseteq D, \quad G \sqcap \exists r.G \sqsubseteq D, \quad B \sqcap \exists r.B \sqsubseteq D \quad \}$$

D steht für „Defekt“

$$q = \exists x\, D(x)$$

$$Q = (q, \mathcal{T})$$

Graph-ABox:

- verwendet keine Konzeptnamen und nur den Rollennamen r
- wenn $r(a,b) \in \mathcal{A}$, dann $r(b,a) \in \mathcal{A}$

Offensichtlich sind Graph-ABoxen dasselbe wie ungerichtete Graphen.

Lemma 7.10

Für alle Graph-ABoxen $\mathcal{A}$ gilt: $\mathcal{A}$ ist nicht 3-färbbar gdw. $\mathcal{A} \models Q$ T7.8

Dies liefert wie gewünscht eine Reduktion von *Nicht*-3-Färbbarkeit auf Beantwortung der Anfrage Q.

Query Rewriting

Ziel

Wir wollen nun Datenbanksysteme verwenden, um OMQs zu beantworten: TBoxen ins relationale Datenbanksystem „schummeln“.

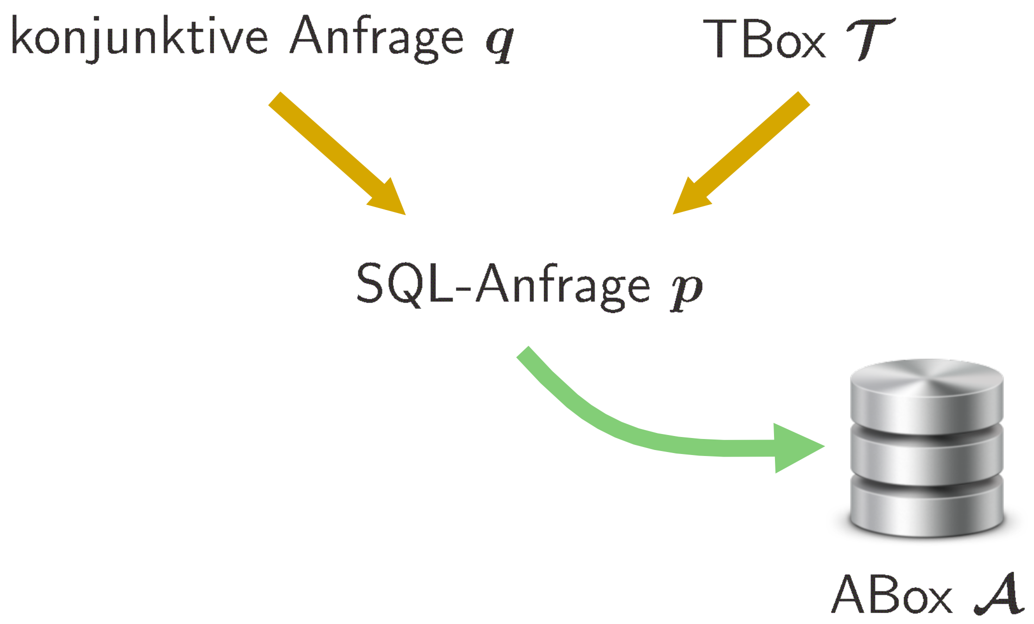
Wir nehmen an:
Die ABox $\mathcal{A}$ ist als Interpretation $\mathcal{I}$ in der Datenbank gespeichert:

$$\begin{aligned}
\Delta^{\mathcal{I}} &= \mathsf{Ind}(\mathcal{A}) \\
A^{\mathcal{I}} &= \{a \mid A(a) \in \mathcal{A}\} \\
r^{\mathcal{I}} &= \{(a,b) \mid r(a,b) \in \mathcal{A}\}
\end{aligned}$$

Wir unterscheiden nicht explizit zwischen diesen beiden Darstellungen; verwenden ABox $\mathcal{A}$ auch als Interpretation.

Query Rewriting

Die Idee von Query Rewriting:



Definition 7.11 [Rewriting]
Sei $Q = (q(\overline{x}), \mathcal{T})$ OMQ.
SQL-Anfrage $p(\overline{x})$ ist *SQL-Rewriting von* Q, wenn für alle ABoxen $\mathcal{A}$ gilt:

$$\text{cert}(Q, \mathcal{A}) = \text{ans}(p, \mathcal{A})$$

ABox als Interpretation

Rewritability: zu schön, um wahr zu sein?

Man kann zeigen, dass Nicht-3-Färbbarkeit *nicht* in SQL ausdrückbar ist.

Es gibt aber viel einfachere OMQs, für die das der Fall ist (TBox in $\mathcal{EL}$):

$$\mathcal{T} = \{\ T \sqsubseteq M,\ \ \exists r.M \sqsubseteq M\ \}$$

$$q = \exists x\,(M(x) \land S(x))$$

Reach-ABox: verwendet nur den Rollennamen r
und enthält Assertionen $S(a)$ und $T(b)$, sonst keine Konzeptassertionen.

Reach-ABoxen sind gerichtete Graphen mit markiertem Start und Ziel.

T7.9

Lemma 7.12

Für alle Reach-ABoxen $\mathcal{A}$ gilt:

$$b \text{ erreichbar von } a \quad \text{gdw.} \quad \mathcal{A} \models Q$$

(Beweis ist recht elementar – probiert es aus.)

Rewritability: zu schön, um wahr zu sein?

Erreichbarkeit ist ebenfalls nicht in SQL ausdrückbar.

Grund ist die *Nicht-Lokalität* dieser Anfrage:
man findet ABoxen $\mathcal{A}_0, \mathcal{A}_1, \ldots,$ die immer größer werden und so dass gilt:

$$\mathcal{A}_i \models Q, \text{ aber } \mathcal{A}' \not\models Q \text{ für alle } \mathcal{A}' \subseteq \mathcal{A}_i$$

$\mathcal{A}_0$ $\qquad S \bullet T$

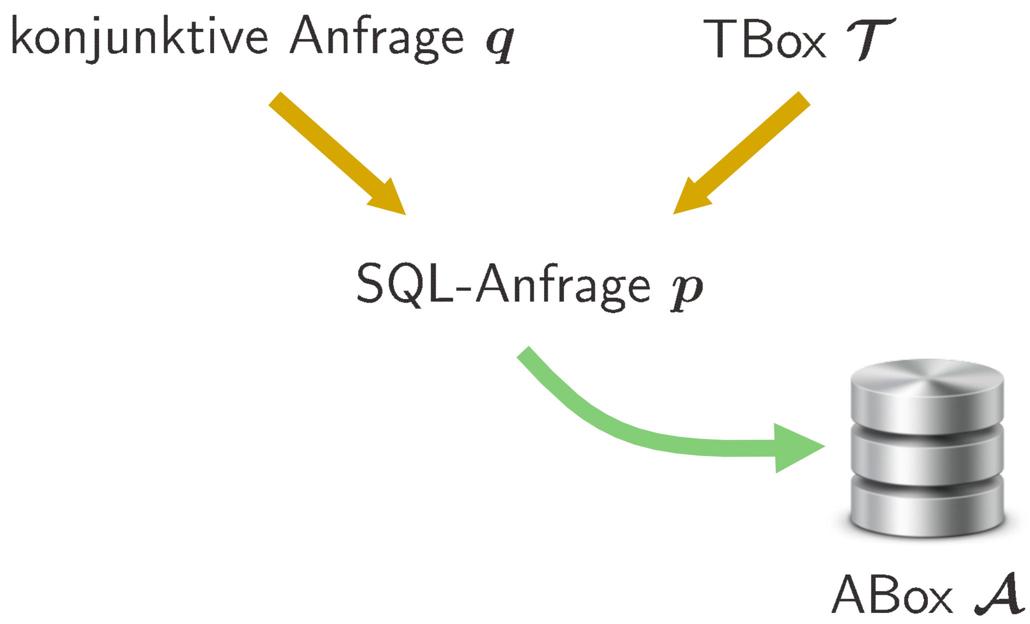
$\mathcal{A}_1$ $\qquad S \bullet \xrightarrow{r} \bullet\, T$

$\mathcal{A}_2$ $\qquad S \bullet \xrightarrow{r} \bullet \xrightarrow{r} \bullet\, T$

(Das bezieht sich aus SQL 1&2. Ab SQL 3 sind einige nicht-lokale Anfragen mittels linearer Rekursion ausdrückbar, zum Beispiel Erreichbarkeit; dennoch ist auch in aktuellen SQL-Versionen nicht jede $\mathcal{EL}$-OMQ ausdrückbar.)

Query Rewriting

Die Idee von Query Rewriting:



Definition 7.11 [Rewriting]

Sei $\boldsymbol{Q} = (\boldsymbol{q}(\overline{\boldsymbol{x}}), \boldsymbol{\mathcal{T}})$ OMQ.

SQL-Anfrage $\boldsymbol{p}(\overline{\boldsymbol{x}})$ ist *SQL-Rewriting von* $\boldsymbol{Q}$, wenn für alle ABoxen $\boldsymbol{\mathcal{A}}$ gilt:

$$\mathsf{cert}(\boldsymbol{Q}, \boldsymbol{\mathcal{A}}) = \mathsf{ans}(\boldsymbol{p}, \boldsymbol{\mathcal{A}})$$

DL-Lite

Definition 7.13 [DL-Lite]

Ein *DL-Lite-Konzept* hat eine der folgenden Formen:

- $\boldsymbol{A}$ (Konzeptname)
- $\top$ (Top-Konzept)
- $\exists \boldsymbol{r}$ (unqualifizierte Existenzrestriktion) kurz für $\exists \boldsymbol{r}.\top$
- $\exists \boldsymbol{r}^-$ (unqualifizierte inverse Existenzrestriktion) kurz für $\exists \boldsymbol{r}^-.\top$

Eine *DL-Lite-TBox* ist eine endliche Menge von

- Konzeptinklusionen $\boldsymbol{C}_1 \sqsubseteq \boldsymbol{C}_2$
- Rolleninklusionen $\boldsymbol{R}_1 \sqsubseteq \boldsymbol{R}_2$ ($\boldsymbol{R}_1, \boldsymbol{R}_2$: Rollennamen $\boldsymbol{r}$ oder Inverse $\boldsymbol{r}^-$)

Wir verzichten hier der Einfachheit haber auf negative Inklusionen

$$\boldsymbol{C}_1 \sqsubseteq \neg \boldsymbol{C}_2 \quad \text{und} \quad \boldsymbol{R}_1 \sqsubseteq \neg \boldsymbol{R}_2$$

die in der ursprünglichen Definition ebenfalls zugelassen sind.

T7.10

DL-Lite

Wir werden sehen:
für konjunktive Anfragen und DL-Lite-TBoxen gibt es stets SQL-Rewritings.

T7.10 cont

Zentrale Technik im Umgang mit DL-Lite: *universelle Modelle*

Hierbei handelt es sich um ein einzelnes(!) Modell,
das genau die „certain answers“ (definiert über alle Modelle) liefert.

Universelle Modelle

Sei $\mathcal{A}$ ABox und $\mathcal{T}$ DL-Lite-TBox. *Universelles Modell* $\mathcal{U}$ für $\mathcal{A}$ und $\mathcal{T}$:

Schritt 1: Starte mit $\mathcal{U}_0$:

$$\begin{aligned} \Delta^{\mathcal{U}_0} &= \mathsf{Ind}(\mathcal{A}) \\ A^{\mathcal{U}_0} &= \{a \mid A(a) \in \mathcal{A}\} \\ r^{\mathcal{U}_0} &= \{(a,b) \mid r(a,b) \in \mathcal{A}\} \end{aligned}$$

Schritt 2: Wende erschöpfend folgende Regeln an:

R1: **Wenn** $d \in C^{\mathcal{U}_i}$, $C \sqsubseteq A \in \mathcal{T}$, $d \notin A^{\mathcal{U}_i}$, **dann** $A^{\mathcal{U}_{i+1}} = A^{\mathcal{U}_i} \cup \{d\}$

R2: **Wenn** $d \in C^{\mathcal{U}_i}$, $C \sqsubseteq \exists R \in \mathcal{T}$, $d \notin (\exists R)^{\mathcal{U}_i}$, **dann** erweitere $\Delta^{\mathcal{U}_i}$ um neues Element e und setze $R^{\mathcal{U}_{i+1}} = R^{\mathcal{U}_i} \cup \{(d,e)\}$.

R3: **Wenn** $(d,e) \in R^{\mathcal{U}_i}$, $R \sqsubseteq S \in \mathcal{T}$, $(d,e) \notin S^{\mathcal{U}_i}$, **dann** $S^{\mathcal{U}_{i+1}} = S^{\mathcal{U}_i} \cup \{(d,e)\}$

(R, S: Rollennamen r oder Inverse r^-)

$\mathcal{U}$ ergibt sich im Limit bei fairer Regelanwendung.

T7.11

Universelle Modelle

Offensichtlich ist $\mathcal{U}$ Modell für $\mathcal{A}$ und $\mathcal{T}$:

- bereits $\mathcal{U}_0$ ist per Definition Modell von $\mathcal{A}$
- da keine Regel mehr anwendbar ist, ist $\mathcal{U}$ Modell von $\mathcal{T}$.

Homomorphismus von Interpretation $\mathcal{I}$ nach Interpretation $\mathcal{J}$
ist Abbildung $h : \Delta^{\mathcal{I}} \to \Delta^{\mathcal{J}}$, so dass:

- $d \in A^{\mathcal{I}}$ impliziert $h(d) \in A^{\mathcal{J}}$
- $(d, e) \in r^{\mathcal{I}}$ impliziert $(h(d), h(e)) \in r^{\mathcal{J}}$

Wir schreiben $\mathcal{I} \to \mathcal{J}$, wenn es Homomorphismus von $\mathcal{I}$ nach $\mathcal{J}$ gibt.

Lemma 7.14
Für jedes Modell $\mathcal{I}$ von $\mathcal{A}$ und $\mathcal{T}$ gilt: $\mathcal{U} \to \mathcal{I}$.

Universelle Modelle

Wir können nun die Haupteigenschaft von universellen Modellen beweisen.

Der Einfachheit halber im Folgenden nur Boolesche Anfragen.

Lemma 7.15

Für jede OMQ $Q = (q, \mathcal{T})$ gilt: $\mathcal{A} \models Q$ gdw. $\mathcal{U} \models q$

Übersetzt auf nicht-Boolesche Anfragen heißt das: $\mathsf{cert}(Q, \mathcal{A}) = \mathsf{ans}(q, \mathcal{U})$

Das universelle Modell repräsentiert also genau die „certain answers“!

Universelle Modelle

Lemma 7.15

Für jede OMQ $Q = (q, \mathcal{T})$ gilt: $\mathcal{A} \models Q$ gdw. $\mathcal{U} \models q$

Beweis: „$\Rightarrow$“.

Wenn $\mathcal{U} \not\models q$, dann $\mathcal{A} \not\models Q$, da $\mathcal{U}$ Modell von $\mathcal{A}$ und $\mathcal{T}$.

„$\Leftarrow$“.

Angenommen $\mathcal{U} \models q$.

Dann gibt es Homomorphismus h von q nach $\mathcal{U}$.

Nach Lemma 7.14 gibt es weiterhin Homomorphismus g
von $\mathcal{U}$ in jedes beliebige Modell $\mathcal{I}$ von $\mathcal{A}$ und $\mathcal{T}$.

Komposition $h(g(\cdot))$ liefert Homomorphismus von q nach $\mathcal{I}$;
also $\mathcal{I} \models q$ für alle Modelle $\mathcal{I}$ von $\mathcal{A}$ und $\mathcal{T}$.

Also $\mathcal{A} \models Q$.

Lokalität

Wesentliche Einsicht: Anfragebeantwortung bzgl. DL-Lite-TBoxen ist *lokal*:

Theorem 7.16

Sei $\boldsymbol{Q} = (\boldsymbol{q}, \boldsymbol{\mathcal{T}})$ OMQ mit DL-Lite-TBox $\boldsymbol{\mathcal{T}}$. Wenn $\boldsymbol{\mathcal{A}} \models \boldsymbol{Q}$, dann gibt es ein $\boldsymbol{\mathcal{A}}' \subseteq \boldsymbol{\mathcal{A}}$ mit $|\mathsf{Ind}(\boldsymbol{\mathcal{A}}')| \leq |\boldsymbol{Q}|^2 + |\boldsymbol{Q}|$, so dass $\boldsymbol{\mathcal{A}}' \models \boldsymbol{Q}$. T7.13

Beachte: die Größe von $\boldsymbol{\mathcal{A}}'$ hängt nur von $\boldsymbol{q}$ und $\boldsymbol{\mathcal{T}}$ ab! T7.12

Definition 7.17

Sei $\boldsymbol{Q} = (\boldsymbol{q}, \boldsymbol{\mathcal{T}})$ OMQ mit DL-Lite-TBox $\boldsymbol{\mathcal{T}}$, $\boldsymbol{n} = |\boldsymbol{Q}|^2 + |\boldsymbol{Q}|$ und $\mathsf{Ind} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{\boldsymbol{n}}\}$ feste Menge von $\boldsymbol{n}$ Individuennamen.

Abox $\boldsymbol{\mathcal{A}}$ ist *kleiner* $\boldsymbol{Q}$*-Zeuge*, wenn

- $\boldsymbol{\mathcal{A}}$ nur Symbole aus $\boldsymbol{Q}$ verwendet;
- $\mathsf{Ind}(\boldsymbol{\mathcal{A}}) \subseteq \mathsf{Ind}$ (also $|\mathsf{Ind}(\boldsymbol{\mathcal{A}})| \leq \boldsymbol{n}$) und
- $\boldsymbol{\mathcal{A}} \models \boldsymbol{Q}$.

Offensichtlich: es gibt nur *endlich viele* kleine $\boldsymbol{Q}$-Zeugen.

Rewritings

Korollar 7.18

Sei $Q = (q, \mathcal{T})$ OMQ mit DL-Lite-TBox $\mathcal{T}$. Dann gilt:

$$\mathcal{A} \models Q \quad \text{gdw.} \quad \text{es gibt kleinen } Q\text{-Zeugen } \mathcal{A}' \text{ mit } \mathcal{A}' \subseteq \mathcal{A}$$

(bis auf Individuenumbenennung)

Lemma 7.19

Für jede ABox $\mathcal{B}$ gibt es Boolesche SQL-Anfrage $q_{\mathcal{B}}$, so dass für alle ABoxen $\mathcal{A}$:

$$\mathcal{B} \subseteq \mathcal{A} \text{ (bis auf Individuenumbenennung)} \quad \text{gdw.} \quad \mathcal{A} \models q_{\mathcal{B}}$$

T7.14

Damit ist die Konstruktion eines SQL-Rewritings offensichtlich:

$$q_{\mathcal{B}_1} \text{ UNION } \cdots \text{ UNION } q_{\mathcal{B}_n}$$

wobei $\mathcal{B}_1, \ldots, \mathcal{B}_n$ alle (endlich vielen) kleinen Q-Zeugen sind.

Zusammenfassung

- Nicht-Lokalität ist der Grund, warum für $\mathcal{EL}$-OMQs und $\mathcal{ALC}$-OMQs im Allgemeinen keine SQL-Rewritings existieren.

- In DL-Lite sind Existenzrestriktionen extrem eingeschränkt: *unquantifiziert*
- Dies resultiert in Lokalität (kleine Q-Zeugen).

- Lokale Anfragen sind in SQL ausdrückbar.
- Hintergrund ist die Äquivalenz von SQL und Prädikatenlogik erster Stufe; für diese ist *Lokalität* eines der wichtigsten Werkzeuge.

Anfragebeantwortung jenseits von DL-Lite

- Wir hatten gesehen: auch für $\mathcal{EL}$-OMQs müssen SQL-Rewritings nicht existieren (Erreichbarkeit).

- Man kann für $\mathcal{EL}$-OMQs aber stets *Datalog*-Rewritings finden; auch dafür gibt es sehr effiziente Systeme.

- Anstatt die TBox in die *Anfrage* zu integrieren, kann man sie in die *ABox* integrieren *(Materialisierung).*

- Materialisierung funktioniert stets für $\mathcal{EL}$; erlaubt auch hier die Verwendung von SQL-Datenbanken.

- Materialisierung funktioniert im Allgemeinen nicht in $\mathcal{ALC}$; in *praktisch relevanten Fällen* aber oftmals doch.

Konjunktive Anfragen mit Ungleichheit

(nur informativ; nicht mehr Bestandteil der VL)

Konjunktive Anfragen mit Ungleichheit

Schon leicht erweiterte Anfragen resultieren in Unentscheidbarkeit.

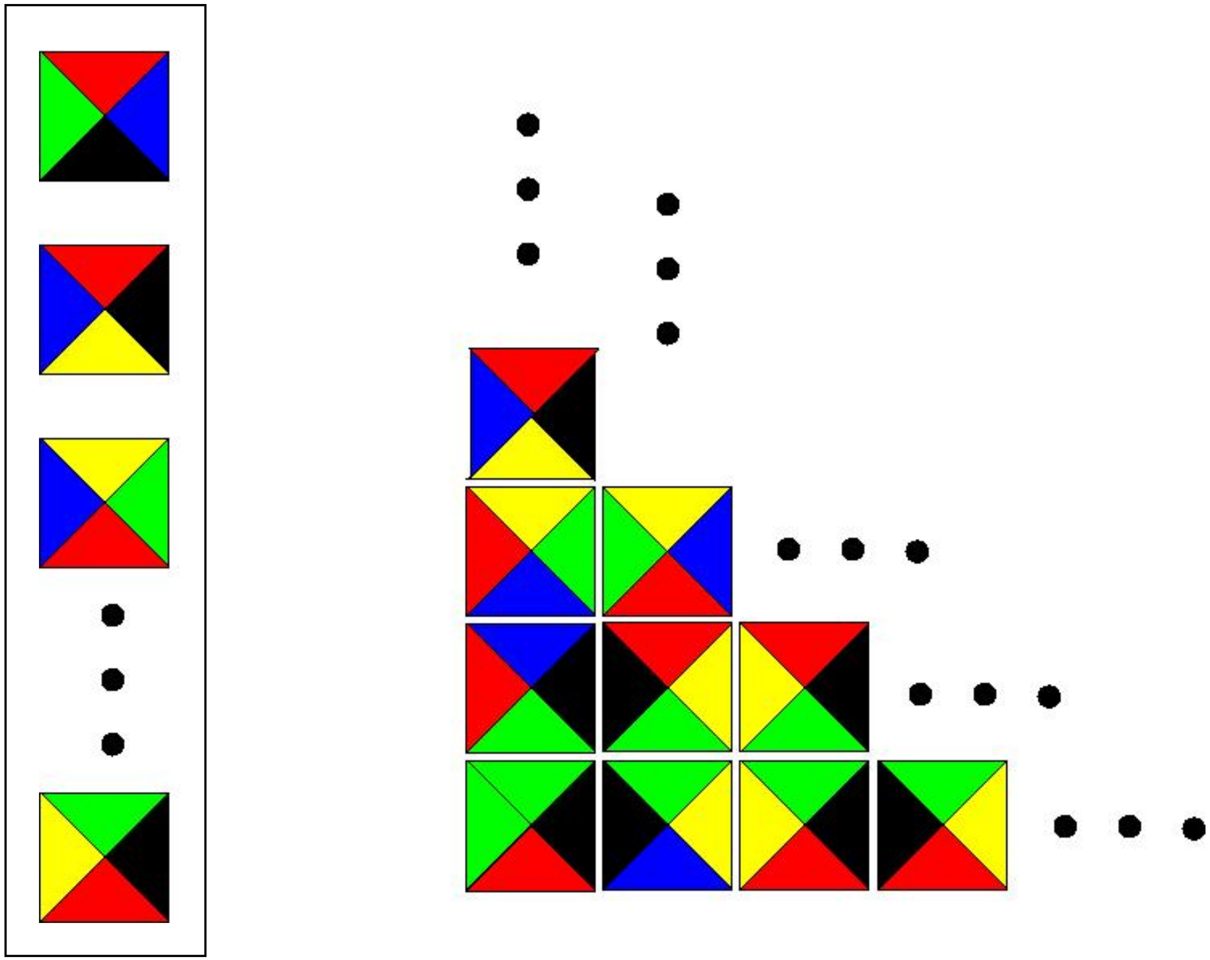
Erweiterte konjunktive Anfrage: erlaubt zusätzlich Atome $x \neq y$

Erweiterte OMQ: entsprechend

Theorem 7.20
Für gegebene ABox $\mathcal{A}$ und erweiterte $\mathcal{ALC}$-OMQ Q ist unentscheidbar, ob $\mathcal{A} \models Q$.

Beweis: Reduktion des unentscheidbaren Dominoproblems.

Domino Problem



Gegeben: endliche Menge von Domino-$Typen$.

Frage: kann man damit den ersten Quadranten der Ebene parkettieren so dass alle benachbarten Kanten dieselbe Farbe haben?

Domino Problem

Definition 7.21 [Domino System]
Domino System ist Paar $\mathcal{D} = (F, T)$ mit

- F endliche Menge von *Farben*;
- T endliche Menge von 4-Tupeln $t = (t_o, t_u, t_\ell, t_r) \in F^4$, den *Domino Typen*

Abbildung $\pi : \mathbb{N} \times \mathbb{N} \to T$ ist *Lösung* für $\mathcal{D}$ wenn

- $\pi(i, j)_r = \pi(i + 1, j)_\ell$ für alle $i, j \geq 0$;
- $\pi(i, j)_o = \pi(i, j + 1)_u$ für alle $i, j \geq 0$;.

Theorem 7.22
Es ist unentscheidbar ob ein gegebenes Dominosystem eine Lösung hat.

Die Reduktion

Ziel:

Gegeben Domino System $\mathcal{D}$, konstruiere erweiterte $\mathcal{ALC}$-OMQ $Q = (q, \mathcal{T})$ so dass $\mathcal{D}$ Lösung hat gdw $\underbrace{\emptyset}_{\text{leere ABox}} \not\models Q$.

Intuition:

Modelle $\mathcal{I}$ von $\mathcal{T}$ so dass $q \not\to \mathcal{I}$ $\approx$ Lösungen für $\mathcal{D}$.

"q-Gegenmodelle"

Signatur von $\mathcal{T}$ und q:

- Rollennamen v, h für vertikale/horizontale Nachbarschaft
- Konzeptname A_t für jedes $t \in T$.

Die Reduktion

1. Jeder Punkt hat horizontalen und vertikalen Nachfolger:

$$\top \sqsubseteq \exists \boldsymbol{h}.\top \sqcap \exists \boldsymbol{v}.\top$$

2. Jeder Punkt hat genau einen Typ:

$$\top \sqsubseteq \bigsqcup_{t \in T} \Big(\boldsymbol{A}_t \sqcap \bigsqcap_{t' \in T, t \neq t'} \neg \boldsymbol{A}_{t'} \Big)$$

3. Benachbarte Kanten haben dieselbe Farbe:

$$\top \sqsubseteq \bigsqcap_{t \in T} \Big(\boldsymbol{A}_t \rightarrow \forall \boldsymbol{h}. \bigsqcup_{t' \in T \textbf{ und } t_r = t'_\ell} \boldsymbol{A}_{t'} \Big)$$

$$\top \sqsubseteq \bigsqcap_{t \in T} \Big(\boldsymbol{A}_t \rightarrow \forall \boldsymbol{v}. \bigsqcup_{t' \in T \textbf{ und } t_o = t'_u} \boldsymbol{A}_{t'} \Big)$$

Die Reduktion

Was noch fehlt:

(∗) jeder hv-Nachfolger ist auch vh-Nachfolger.

Wir erreichen das mittels q:

$$\exists y_0 \exists y_1 \exists y_2 \exists y_3 \exists y_2' \, h(y_0, y_1) \land v(y_1, y_2) \land \\ v(y_0, y_3) \land h(y_3, y_2') \land y_2 \neq y_2'$$

Lemma 7.23
$\mathcal{D}$ hat Lösung gdw. $\emptyset \not\models Q$.

Beachte: für Korrektheit müssen h und v nicht unbedingt Funktionen sein

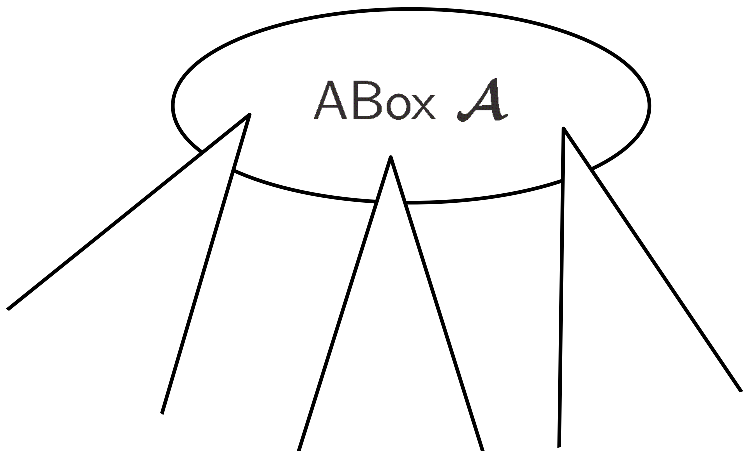
Theorem 7.20 folgt unmittelbar.

Nachbemerkung

Wie konnte so plötzlich aus entscheidbarem Problem unentscheidbares werden?

Auch für das Beantworten (nicht-erweiterter) konjunktiver Anfragen gibt es eine Art Baummodelleigenschaft.

Wenn $\mathcal{A} \not\models Q$, dann gibt es immer ein Gegenmodell der Form



beliebige Struktur

unendliche Bäume

Die Konstruktion des Gitters zeigt:

Mit Ungleichheit in der Anfrage gibt es keine baumförmigen Gegenmodelle mehr!

Nachbemerkungen zur Vorlesung

Auslassungen

Viele relevante Themen ausgelassen:

- „eingebaute“ Datentypen (Zahlen, Strings, etc.)
  zur Repräsentation von z. B. Alter, Gewicht, Distanz, etc.
- Zusammenhang zur Automatentheorie

★ Modularität von TBoxen, Modulextraktion

★ Erklärungen von Inferenzen (Subsumtionen, Unerfüllbarkeit)

★ Erweiterungen von DLs: z. B. temporale DLs, probabilistische DLs

- Entwurfsmethodologien für TBoxen
- Anwendungen wie das Semantische Web

★ Aktuelle Forschung, mehr dazu morgen ★ Aktuelle Forschung

OWL

Vom W3C (World Wide Web Committee) standardisierte Ontologiesprache:

- soll im semantischen Web Verwendung finden
- basiert auf Beschreibungslogik
- hat XML-Syntax, kann in RDFS (Resource Description Framework) übersetzt werden:

Zwei Versionen von OWL:

- OWL 1.0 (W3C-Standard 2004)
- OWL 2.0 (W3C-Standard 2009)

OWL

OWL 1.0: $\mathcal{ALC}$ plus

- Zahlenrestriktionen
- Inverse Rollen
- Transitive Rollen
- Nominale (Konzepte mit genau einer Instanz in jedem Modell)
- etc.

Erfüllbarkeit ist entscheidbar und NExpTime-vollständig.

OWL

OWL 2.0: OWL 1.0 plus

- reflexive und symmetrische Rollen
- Rolleninklusionen $r_1 \circ r_2 \sqsubseteq s$
- etc.

Erfüllbarkeit ist $2\textsc{NExpTime}$-vollständig.

Wegen hoher Komplexität:
drei offizielle „Profile“ mit besseren Berechnungseigenschaften, z. B.

- Schlussfolgern in Polynomialzeit
  (z. B. Profile EL, QL: basierend auf $\mathcal{EL}$ bzw. DL-Lite)
- effiziente Beantwortung konjunktiver Anfragen mit Standard-DB-Systemen
  (Profil QL)

Danke für's Teilnehmen!

Morgen:

- Aktuelle Forschung:
  Überblick zum Thema Modularität von Ontologien
- kurze Auswertung der Evaluationsergebnisse
- kurze Ansagen zu Fachgesprächen/Prüfungen
- Werbung für Wahlveranstaltung im Wintersemester

Literatur für dieses Kapitel

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

An Introduction to Description Logic.

Cambridge University Press, 2017.

Kapitel 7: Query answering.

Relevant sind §7.1 und §7.2;
für einen Überblick empfehle ich auch §7.4 und §7.5.

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.