

## Konjunktive Anfragen und Beschreibungslogik-TBoxen

23

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)

24

# Ontology-Mediated Queries

## Definition 7.9 (OMQ)

*Ontology-mediated query (OMQ)* ist Paar  $Q = (q(\bar{x}), \mathcal{T})$

mit  $q(\bar{x})$  konjunktiver Anfrage und  $\mathcal{T}$  TBox.

*Antwort auf  $Q$  in ABox  $\mathcal{A}$* : Tupel  $\bar{a} = a_1 \cdots a_n$  von Individuen, das Antwort auf  $q(\bar{x})$  in *allen* Modellen von  $\mathcal{A}$  und  $\mathcal{T}$  ist.

Menge aller Antworten auf  $Q$  in  $\mathcal{A}$  bezeichnen wir mit  $\text{cert}(Q, \mathcal{A})$ .

Solche Antworten nennt man auch *sichere Antworten* (engl.: certain answers).

Beachte:

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

T7.7

25

# Beantwortung von OMQs

*Anfragebeantwortung:*

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

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

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

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

Für Boolesche OMQs ist Anfragebeantwortung also ein *Entscheidungsproblem*.

26

## 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  $f : V \rightarrow \{R, G, B\}$  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  $\mathcal{ALC}$ ):

$$\mathcal{T} = \left\{ \begin{array}{l} \top \sqsubseteq R \sqcup G \sqcup B, \quad R \sqcap \exists r.R \sqsubseteq D, \\ \quad \quad \quad \quad \quad \quad \quad G \sqcap \exists r.G \sqsubseteq D, \\ \quad \quad \quad \quad \quad \quad \quad B \sqcap \exists r.B \sqsubseteq D \end{array} \right\} \quad \begin{array}{l} D \text{ steht für} \\ \text{„Defekt“} \end{array}$$

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

27

## Datenkomplexität von OMQs

$$\mathcal{T} = \left\{ \begin{array}{l} \top \sqsubseteq R \sqcup G \sqcup B, \quad R \sqcap \exists r.R \sqsubseteq D, \\ \quad \quad \quad \quad \quad \quad \quad G \sqcap \exists r.G \sqsubseteq D, \\ \quad \quad \quad \quad \quad \quad \quad B \sqcap \exists r.B \sqsubseteq D \end{array} \right\} \quad \begin{array}{l} D \text{ steht für} \\ \text{„Defekt“} \end{array}$$

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

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

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  $Q$ .

28

## Kapitel 7

### 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:

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

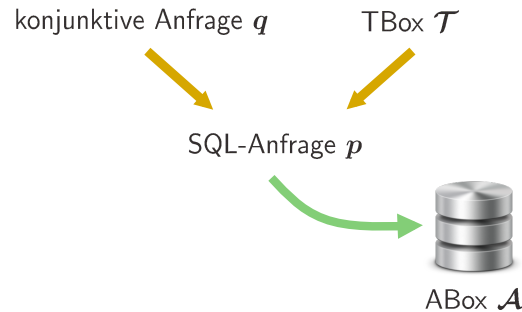
Wir unterscheiden nicht explizit zwischen diesen beiden Darstellungen; verwenden ABox  $\mathcal{A}$  auch als Interpretation.

29

30

## Query Rewriting

Die Idee von Query Rewriting:



**Definition 7.11** [Rewriting]

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

SQL-Anfrage  $p(\bar{x})$  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

31

## 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  $\mathcal{EL}$ ):

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

$$q = \exists x (M(x) \wedge S(x))$$

*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.

**Lemma 7.12**

T7.9

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

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

(Beweis ist recht elementar – probiert es aus.)

32

## Rewritability: zu schön, um wahr zu sein?

Erreichbarkeit ist ebenfalls nicht in SQL ausdrückbar.

Grund ist die *Nicht-Lokalität* dieser Anfrage:

man findet ABoxen  $\mathcal{A}_0, \mathcal{A}_1, \dots$ , die immer größer werden und so dass gilt:

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

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

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

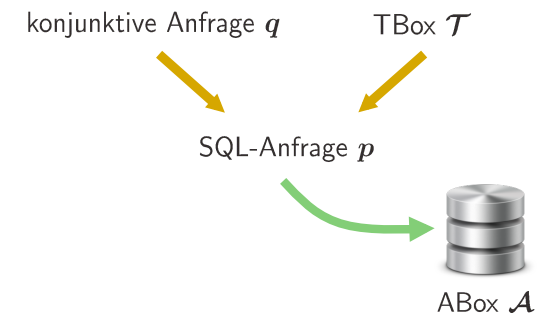
$$\mathcal{A}_2 \quad S \xrightarrow{r} \bullet \xrightarrow{r} \bullet T$$

(Das bezieht sich auf 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.)

33

## Query Rewriting

Die Idee von Query Rewriting:



**Definition 7.11** [Rewriting]

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

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

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

34

### Definition 7.13 [DL-Lite]

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

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

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

- Konzeptinklusionen  $C_1 \sqsubseteq C_2$
- Rolleninklusionen  $R_1 \sqsubseteq R_2$       ( $R_1, R_2$ : Rollennamen  $r$  oder Inverse  $r^-$ )

Wir verzichten hier der Einfachheit halber auf negative Inklusionen

$$C_1 \sqsubseteq \neg C_2 \text{ und } R_1 \sqsubseteq \neg R_2$$

die in der ursprünglichen Definition ebenfalls zugelassen sind.

T7.10

35

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  $\mathcal{U}_0$ :

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

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

37

## 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  $h : \Delta^{\mathcal{I}} \rightarrow \Delta^{\mathcal{J}}$ , 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  $\mathcal{I} \rightarrow \mathcal{J}$ , 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:  $\mathcal{U} \rightarrow \mathcal{I}$ .



## 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:  $\text{cert}(Q, \mathcal{A}) = \text{ans}(q, \mathcal{U})$

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

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

### Definition 7.17

Sei  $Q = (q, \mathcal{T})$  OMQ mit DL-Lite-TBox  $\mathcal{T}$ ,  $n = |Q|^2 + |Q|$  und  $\text{Ind} = \{a_1, \dots, a_n\}$  feste Menge von  $n$  Individuennamen.

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

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

Offensichtlich: es gibt nur *endlich viele* kleine  $Q$ -Zeugen.

## Rewritings

### Korollar 7.18

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

$\mathcal{A} \models Q$  gdw. es gibt kleinen  $Q$ -Zeugen  $\mathcal{A}'$  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  $\mathcal{A}$ :

$\mathcal{B} \subseteq \mathcal{A}$  (bis auf Individuenumbenennung) gdw.  $\mathcal{A} \models q_{\mathcal{B}}$  T7.14

Damit ist die Konstruktion eines SQL-Rewritings offensichtlich:

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

wobei  $\mathcal{B}_1, \dots, \mathcal{B}_n$  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.

## Kapitel 7

### Konjunktive Anfragen mit Ungleichheit

(nur informativ; nicht mehr Bestandteil der VL)

## 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

Schon leicht erweiterte Anfragen resultieren in Unentscheidbarkeit.

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

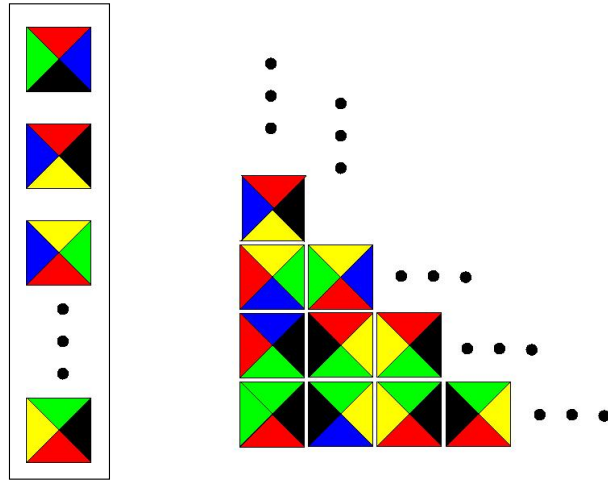
*Erweiterte* OMQ: entsprechend

**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?

47

## 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  $\pi : \mathbb{N} \times \mathbb{N} \rightarrow T$  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.

48

## Die Reduktion

Ziel:

Gegeben Domino System  $\mathcal{D}$ , konstruiere erweiterte  $\mathcal{ALC}$ -OMQ  $Q = (q, \mathcal{T})$  so dass  $\mathcal{D}$  Lösung hat gdw  $\emptyset \neq Q$ .

|  
leere ABox

Intuition:

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

“ $q$ -Gegenmodelle”

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

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

49

## Die Reduktion

1. Jeder Punkt hat horizontalen und vertikalen Nachfolger:

$$\top \sqsubseteq \exists h. \top \sqcap \exists v. \top$$

2. Jeder Punkt hat genau einen Typ:

$$\top \sqsubseteq \bigsqcup_{t \in T} (A_t \sqcap \prod_{t' \in T, t' \neq t} \neg A_{t'})$$

3. Benachbarte Kanten haben dieselbe Farbe:

$$\top \sqsubseteq \prod_{t \in T} (A_t \rightarrow \forall h. \bigsqcup_{t' \in T \text{ und } t_r = t'_\ell} A_{t'})$$

$$\top \sqsubseteq \prod_{t \in T} (A_t \rightarrow \forall v. \bigsqcup_{t' \in T \text{ und } t_o = t'_u} A_{t'})$$

50

## Die Reduktion

Was noch fehlt:

(\*) jeder  $hv$ -Nachfolger ist auch  $vh$ -Nachfolger.

Wir erreichen das mittels  $q$ :

$$\exists y_0 \exists y_1 \exists y_2 \exists y_3 \exists y'_2 \ h(y_0, y_1) \wedge v(y_1, y_2) \wedge \\ v(y_0, y_3) \wedge h(y_3, y'_2) \wedge y_2 \neq y'_2$$

### 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.

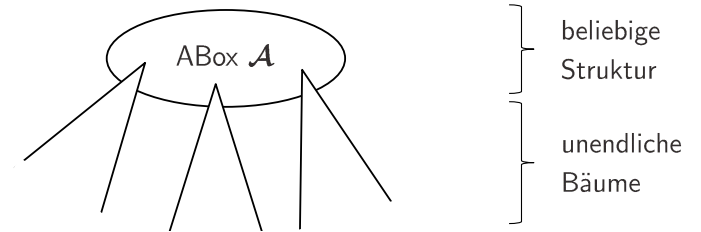
51

## 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



Die Konstruktion des Gitters zeigt:

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

52

## 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

53

54

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)

55

OWL 1.0: *ALC* plus

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

Erfüllbarkeit ist entscheidbar und  $\text{NEXPTIME}$ -vollständig.

56

OWL 2.0: OWL 1.0 plus

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

Erfüllbarkeit ist  $2\text{NEXPTIME}$ -vollständig.

Wegen hoher Komplexität:

drei offizielle „Profile“ mit besseren Berechnungseigenschaften, z. B.

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

57

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

58

## 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.