

Struktur Vorlesung

- Kapitel 1: Einleitung
- Kapitel 2: Grundlagen
- Kapitel 3: Ausdrucksstärke und Modellkonstruktionen
- Kapitel 4: Tableau Algorithmen
- Kapitel 5: Komplexität
- Kapitel 6: ABoxen und Anfragebeantwortung
- Kapitel 7: Effiziente Beschreibungslogiken

ABoxen und Anfragebeantwortung

Ziel des Kapitels

TBoxen repräsentieren allgemeines, begriffliches Wissen

Um *konkrete Situationen* zu repräsentieren, braucht man *Instanzen*.

Für medizinische Ontologien wie SNOMED z.B. Patientendaten:

Patient(p_1)	Patient(p_2)
Medikament(m)	Krankheit(k)
erhält(p_1, m)	erhält(p_2, m)
heilt(m, k)	hat(p_1, k)

Patient, Medikament, etc können in TBox definiert sein.

Ziel des Kapitels

Ziel des Kapitels:

- Einführen eines Formalismus für Instanzdaten (ABox)
- Auswahl Schlußfolgerungsprobleme, um mit Instanzdaten zu arbeiten (insb. verschiedene Varianten von Anfragebeantwortung)
- Algorithmen entwickeln / Komplexität analysieren.

ABoxen und Anfragebeantwortung

Grundlagen

ABox - Syntax

Wir reservieren eine unendliche Menge von *Individuen* a, b, \dots
Diese entsprechen Konstanten im Sinne der Prädikatenlogik

Definition 6.1 (ABox Syntax)

Eine

- *Konzeptassertion* hat die Form $C(a)$
- *Rollenassertion* hat die Form $r(a, b)$

Eine *ABox* ist eine endliche Menge von (Konzept- und Rollen-)assertionen.

T6.1

ABox – Semantik

Definition 6.1 (ABox Semantik)

Interpretation \mathcal{I}

- erfüllt $C(a)$ gdw. $a \in C^{\mathcal{I}}$;
- erfüllt $r(a, b)$ gdw. $(a, b) \in r^{\mathcal{I}}$;

\mathcal{I} ist *Modell* von \mathcal{A} gdw. \mathcal{I} alle Assertionen in \mathcal{A} erfüllt.

T6.1 cont

Beachte:

Modell \mathcal{I} darf zusätzlich Assertionen wahr machen, die in \mathcal{A} *nicht* vorkommen

Das in ABoxen repräsentierte Wissen ist also *unvollständiges Wissen*
(und dasselbe gilt natürlich auch für TBoxen)

Wissensbasis

Definition 6.2 (Wissensbasis)

Wissensbasis (WB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ besteht aus TBox \mathcal{T} und ABox \mathcal{A} .

Interpretation \mathcal{I} ist *Modell* von \mathcal{K} wenn \mathcal{I} Modell von \mathcal{T} und von \mathcal{A} .

In Anwendungen haben \mathcal{T} und \mathcal{A} verschiedenen Status:

- \mathcal{T} wird einmal erstellt, ändert sich danach üblicherweise nicht mehr
- \mathcal{A} ändert sich häufig (wie Datenbank)

ABoxen und Anfragebeantwortung

Schlussfolgerungsprobleme

Grundlegende Schlussfolgerungsprobleme

Definition 6.3 (Konsistenz, Instanz)

Sei \mathcal{K} Wissensbasis, $C(a)$ Konzeptassertion. Dann ist

- \mathcal{K} *konsistent* wenn \mathcal{K} Modell hat;
- a eine *Instanz* von C bzgl. \mathcal{K} wenn jedes Modell von \mathcal{K} auch $C(a)$ erfüllt. Wir schreiben dann $\mathcal{K} \models C(a)$.

T6.2

Grundlegende Schlussfolgerungsprobleme

Konsistenzproblem:

gegeben \mathcal{K} , entscheide ob \mathcal{K} konsistent;

Instanzproblem:

gegeben \mathcal{K} und $C(a)$, entscheide ob $\mathcal{K} \models C(a)$;

Reduktionen

Konsistenz- und (Nicht-)Instanzproblem wechselseitig polynomiell reduzierbar.

Lemma 6.4.

- \mathcal{K} konsistent gdw. $\mathcal{K} \not\models \perp(a)$
- $(\mathcal{T}, \mathcal{A}) \models C(a)$ gdw. $(\mathcal{T}, \mathcal{A} \cup \{\neg C(a)\})$ inkonsistent.

T6.3

Erfüllbarkeit (von Konzepten) polynomiell reduzierbar auf Konsistenz.

Lemma 6.5.

C erfüllbar bzgl. \mathcal{T} gdw. $(\mathcal{T}, \{C(a)\})$ konsistent.

Intuitiv:

Erfüllbarkeit entspricht genau KB Konsistenz mit ABoxen der Form $\{C(a)\}$

Anfragebeantwortung

Viele Anwendungen verwenden ABoxen wie (semantische) Datenbanken

Verschiedene Anfragesprachen möglich:

- *Instanzanfrage*: gegeben \mathcal{K} und C , ermittle alle a mit $\mathcal{K} \models C(a)$
- *Konjunktive Anfragen*: generalisieren Instanzanfragen, Definition später.

Berechnungsproblem, kein Entscheidungsproblem.

T6.2cont

ABoxen vs Relationale Datenbanken

Relationale Datenbanken, zur Erinnerung:

- *relationales Schema* ist Menge von Tabellennamen mit Stelligkeit
- konkrete *Datenbankinstanz* füllt die Tabellen mit Inhalten
- Anfragen sind in SQL formuliert

Einfache ABox: in jeder Konzeptassertion $C(a)$ ist C ein Konzeptname.

Im Prinzip mögliche Betrachtungsweise:

- einfache ABox kann als Datenbankinstanz dargestellt werden
- Instanzanfrage kann (meist) als SQL-Anfrage dargestellt werden **T6.4**

Es gibt jedoch einen **drastischen Unterschied in der Semantik!**

ABoxen vs Relationale Datenbanken

Relationale Datenbanken machen die Closed World Assumption (CWA):
was nicht explizit als wahr angegeben ist, ist falsch.

Für ABoxen machen wir die Open World Assumption (OWA):
es sind nur solche Dinge falsch, die explizit als falsch angegeben sind

T6.4cont

Beachte:
die OWA ist notwendig, damit die Verwendung einer TBox Sinn macht!

T6.4cont

ABoxen vs Datenbanken

Der OWA/CWA Unterschied hat sehr weitreichende Konsequenzen:

- Anfragebeantwortung in relationalen Datenbanken
entspricht dem Model Checking Problem: Datenbank = Modell
Anfrage = logische Formel
- Anfragebeantwortung in Beschreibungslogik
entspricht logischer Folgerbarkeit: KB = logische Theorie
Anfrage = logische Formel

Letzteres ist in der Regel ein **wesentlich schwierigeres Problem**

ABoxen und Anfragebeantwortung

Instanzanfragen

Übersicht

Instanzanfrage C and Wissensbasis $\mathcal{K} = (\mathcal{T}, \mathcal{A})$:

- berechenbar durch mehrfaches Entscheiden des Instanzproblems:
überprüfe ob $\mathcal{K} \models C(a)$ für all Individuen a in \mathcal{A}
- Instanzproblem kann auf Konsistenzproblem reduziert werden

Wir konzentrieren uns also auf das Entscheiden von Konsistenz
(in der Praxis ist das allerdings nicht sehr effizient)

Mögliche Algorithmen für Konsistenz:

- Erweiterung von *ALC*-Elim (bzw. *ALC*-Worlds wenn $\mathcal{T} = \emptyset$)
- Erweiterung von Tableau Algorithmen
- Reduktion auf Erfüllbarkeit von Konzepten bzgl. \mathcal{T}

Vervollständigung

Vervollständigung (Precompletion):

Turing-Reduktion der Konsistenz von Wissensbasen $\mathcal{K} = (\mathcal{T}, \mathcal{A})$
auf die Erfüllbarkeit gewisser Konzepte bzgl. \mathcal{T}

Notation: $\text{Ind}(\mathcal{A})$ ist Menge aller Individuen in \mathcal{A}

Grundideen:

- Zerlege ABox in Teile: ein Teil pro Individuum $a \in \text{Ind}(\mathcal{A})$.
- Für jedes $a \in \text{Ind}(\mathcal{A})$, konstruiere Konzept C_a und teste ob C_a erfüllbar bzgl. \mathcal{T}
- Aus den gesammelten Modellen kann man Modell von \mathcal{K} konstruieren.

- Grundidee: $C_a = \bigsqcap_{D(a) \in \mathcal{A}} D$ (funktioniert so aber noch nicht) T6.5

Vervollständigung

Bezüglich der Eingabe $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ nehmen wir an:

- Alle Konzepte in \mathcal{A} sind in NNF
- \mathcal{T} hat die Form $\{\top \sqsubseteq C_{\mathcal{T}}\}$ mit $C_{\mathcal{T}}$ in NNF

Definition 6.6. (Teilkonzepte von ABox und Wissensbasis)

Für alle ABoxen \mathcal{A} und Wissensbasen $\mathcal{K} = (\mathcal{T}, \mathcal{A})$:

- $\text{sub}(\mathcal{A}) = \bigcup_{C(a) \in \mathcal{A}} \text{sub}(C)$
- $\text{sub}(\mathcal{K}) = \text{sub}(\mathcal{T}) \cup \text{sub}(\mathcal{A})$

Vervollständigung

Definition 6.7. (Vervollständigung)

Vervollständigung von \mathcal{ALC} -Wissensbasis $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ ist Wissensbasis $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ so dass:

1. \mathcal{A}' erweitert \mathcal{A} um Assertionen der Form $C(a)$ mit $C \in \text{sub}(\mathcal{K})$ und $a \in \text{Ind}(\mathcal{A})$
2. $C_{\mathcal{T}}(a) \in \mathcal{A}'$ für alle $a \in \text{Ind}(\mathcal{A})$
3. wenn $C \sqcap D(a) \in \mathcal{A}'$, dann $C(a) \in \mathcal{A}'$ und $D(a) \in \mathcal{A}'$
4. wenn $C \sqcup D(a) \in \mathcal{A}'$, dann $C(a) \in \mathcal{A}'$ oder $D(a) \in \mathcal{A}'$
5. wenn $\forall r. C(a) \in \mathcal{A}'$ und $r(a, b) \in \mathcal{A}'$, dann $C(b) \in \mathcal{A}'$

T6.5 cont

Vervollständigung

Lemma 6.8.

\mathcal{K} konsistent gdw. es gibt Vervollständigung $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ von \mathcal{K} so dass

$C_a := \bigsqcap_{C(a) \in \mathcal{A}'} C$ erfüllbar bzgl. \mathcal{T} für alle $a \in \text{Ind}(\mathcal{A}')$. T6.7

Es gibt offensichtlich nur endlich viele Vervollständigungen

Theorem 6.9.

ABox Konsistenz in \mathcal{ALC} ist entscheidbar. T6.8

Wir haben damit auch einen Algorithmus für Anfragebeantwortung.

Komplexität

Mit wenigen Ausnahmen:

Konsistenzproblem hat selbe Komplexität wie Erfüllbarkeit

Für *ALC* also zu erwarten:

- EXPTIME-vollständig mit TBoxen;
- PSPACE-vollständig ohne TBoxen.

Untere Schranken: Kapitel 5 + Lemma 6.5

Obere Schranken: im folgenden aus Kapitel 5 + Lemma 6.8

Resultat

Definition 6.10.

Größe $|\mathcal{A}|$ einer ABox \mathcal{A} ist

$$\sum_{C(a) \in \mathcal{A}} |C| + 3 + 6 \cdot |\{r(a, b) \in \mathcal{A}\}|$$

Größe $|\mathcal{K}|$ einer WB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ ist $|\mathcal{T}| + |\mathcal{A}|$.

Lemma 6.11.

Für jede WB \mathcal{K} gibt es höchstens $2^{\mathcal{O}(n^2)}$ Vervollständigungen, $n = |\mathcal{K}|$.

T6.9

Theorem 6.12.

Das Konsistenzproblem in \mathcal{ALC} ist

1. EXPTIME-vollständig mit generellen TBoxen;
2. PSPACE-vollständig ohne TBoxen.

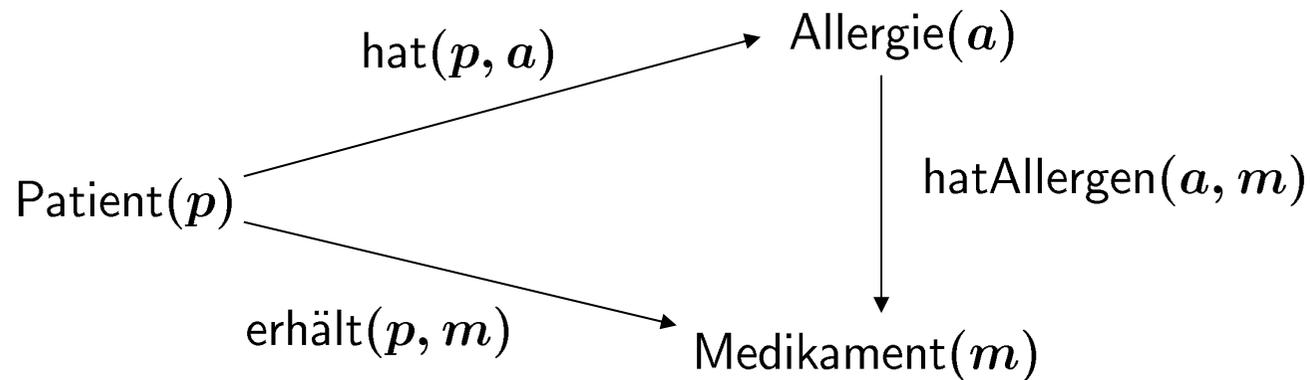
T6.10

ABoxen und Anfragebeantwortung

Konjunktive Anfragen

Instanzanfragen und ABoxen

ABoxen können komplexe relationale Strukturen beschreiben:



Konzept kann solche Struktur (mit p als Wurzel) nicht “abfragen”

(denn die Struktur ist nicht baumförmig, siehe Kapitel 3 + Unravelling)

Konjunktive Anfragen

Konjunktive Anfragen

- generalisieren Instanzanfragen in natürlicher Weise;
- erlauben Anfragen bzgl. der *relationalen Struktur* von ABoxen;
- entsprechen einem wichtigen Fragment von SQL;
- sind in der Datenbankwelt sehr weit verbreitet.

Konjunktive Anfrage - Syntax

Von nun an sei \mathbf{N}_V eine Menge von *Variablen*.

Definition 6.13. (Konjunktive Anfrage)

Ein

- *Konzeptatom* hat die Form $C(x)$, mit $x \in \mathbf{N}_V$;
- *Rollenatom* hat die Form $r(x, y)$, mit $x, y \in \mathbf{N}_V$;

Eine *Konjunktive Anfrage* hat die Form $\exists \bar{y} \varphi(\bar{x}, \bar{y})$ wobei

- $\bar{y} = y_0 \cdots y_m$ die *quantifizierten Variablen*
- $\bar{x} = x_0 \cdots x_m$ die *Antwortvariablen*
- $\varphi(\bar{x}, \bar{y})$ Konjunktion von Konzept- und Rollenatomen über den Variablen $\bar{x} \cup \bar{y}$

T6.11

Konjunktive Anfrage - Notation

Wir schreiben

- x, y, z für Variablen
- $\bar{x}, \bar{y}, \bar{z}$ für Tupel von Variablen
- $\varphi(\bar{x}, \bar{y})$ für Konjunktionen von Atomen über Variablen $\bar{x} \cup \bar{y}$
- $q(\bar{x})$ für konjunktive Anfragen mit Antwortvariablen \bar{x}

Konjunktive Anfrage - Semantik

Definition 6.14. (Homomorphismus, Antwort bzgl. Interpretation)

Sei \mathcal{I} Interpretation und $q(\bar{x}) = \exists \bar{y} \varphi(\bar{x}, \bar{y})$ mit $\bar{x} = x_1 \cdots x_n$.

Abbildung $\tau : \bar{x} \cup \bar{y} \rightarrow \Delta^{\mathcal{I}}$ ist *Homomorphismus* von $q(\bar{x})$ nach \mathcal{I} wenn

- $\tau(x) \in C^{\mathcal{I}}$ für alle Konzeptatome $C(x)$ in φ ;
- $(\tau(x), \tau(y)) \in r^{\mathcal{I}}$ für alle Rollenatome $r(x, y)$ in φ ;

Antwort auf $q(\bar{x})$ bzgl. \mathcal{I} : Tupel $a_1 \cdots a_n$ von Individuen so dass es Homomorphismus τ von $q(\bar{x})$ nach \mathcal{I} gibt mit $\tau(x_i) = a_i$ für $1 \leq i \leq n$.

T6.12

Konjunktive Anfrage - Semantik

Definition 6.15. (Antwort bzgl. Wissensbasis)

Sei $q(\bar{x}) = \exists \bar{y} \varphi(\bar{x}, \bar{y})$ mit $\bar{x} = x_1 \cdots x_n$ und \mathcal{K} Wissensbasis.

Antwort auf $q(\bar{x})$ bzgl. \mathcal{K} : Tupel

$$a_1 \cdots a_n$$

das Antwort auf $q(\bar{x})$ ist bzgl. aller Modelle \mathcal{I} von \mathcal{K} .

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

Beantwortung konjunktiver Anfragen: gegeben $q(\bar{x})$ und \mathcal{K} , berechne
alle Antworten auf $q(\bar{x})$ bzgl. \mathcal{K}

T6.13

Konjunktive Anfragen vs. Instanzanfragen

Konjunktive Anfrage $\exists \bar{y} \varphi(\bar{x}, \bar{y})$ kann als knoten- und kantenbeschrifteter gerichteter Graph (V, E, \mathcal{L}) gesehen werden:

- $V = \bar{x} \cup \bar{y}$;
- $(x, r, y) \in E$ wenn $r(x, y)$ Atom in $q(\bar{x})$ ist;
- $\mathcal{L}(x) = \{C \mid C(x)$ ist Atom in $q(\bar{x})\}$.

Intuitiv:

Instanzanfragen = baumförmige konjunktive Anfragen.

mit Wurzel einziger Antwortvariable

T6.14

Konjunktive Anfragen und SQL

Datenbankwelt:

Konjunktive Anfragen entsprechen *select-project-join*-Anfragen, eine eingeschränkte aber sehr wichtige Art von SQL-Anfrage

In Datenbanksystemen

- sind $>90\%$ aller gestellten Anfragen *select-project-join*;
- sind die query engines speziell auf solche Anfragen hin optimiert.

Man kann zeigen, dass die Beantwortung beliebiger SQL-Anfragen bzgl. *ACC*-Wissensbasen unentscheidbar ist!

Beantwortung konjunktiver Anfragen

- Reduktion auf mehrere Konsistenztests, wie auch bei der Beantwortung von Instanzanfragen
- Die Reduktion ist technisch aber viel aufwendiger und es sind exponentiell viele Konsistenztests notwendig.
- Im Fall von *ALC* erhält man trotzdem ein EXPTIME Verfahren
- Leichte Erweiterungen der BL können das aber ändern, z.B. im Fall von *ALCI*:
 - Beantworten von Instanzanfragen immernoch EXPTIME -vollständig
 - Beantworten von Konjunktiven Anfragen 2EXPTIME -vollständig

Konjunktive Anfragen mit Ungleichheit

Schon leichte Erweiterungen resultieren in Unentscheidbarkeit.

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

Wir betrachten der Einfachheit halber *Boolsche (erweiterte) Anfragen*
d.h. Anfragen ohne Antwortvariablen

Boolsche Anfrage q liefert bzgl. jeder Wissensbasis \mathcal{K} entweder

- keine Antwort
(es gibt Modell \mathcal{I} von \mathcal{K} , so dass kein Hom. von q nach \mathcal{I} existiert)
- das leere Tupel $()$ als einzige Antwort; wir schreiben dann $\mathcal{K} \models q$
(für alle Modelle \mathcal{I} von \mathcal{K} gilt: es gibt Hom. von q nach \mathcal{I})

T6.15

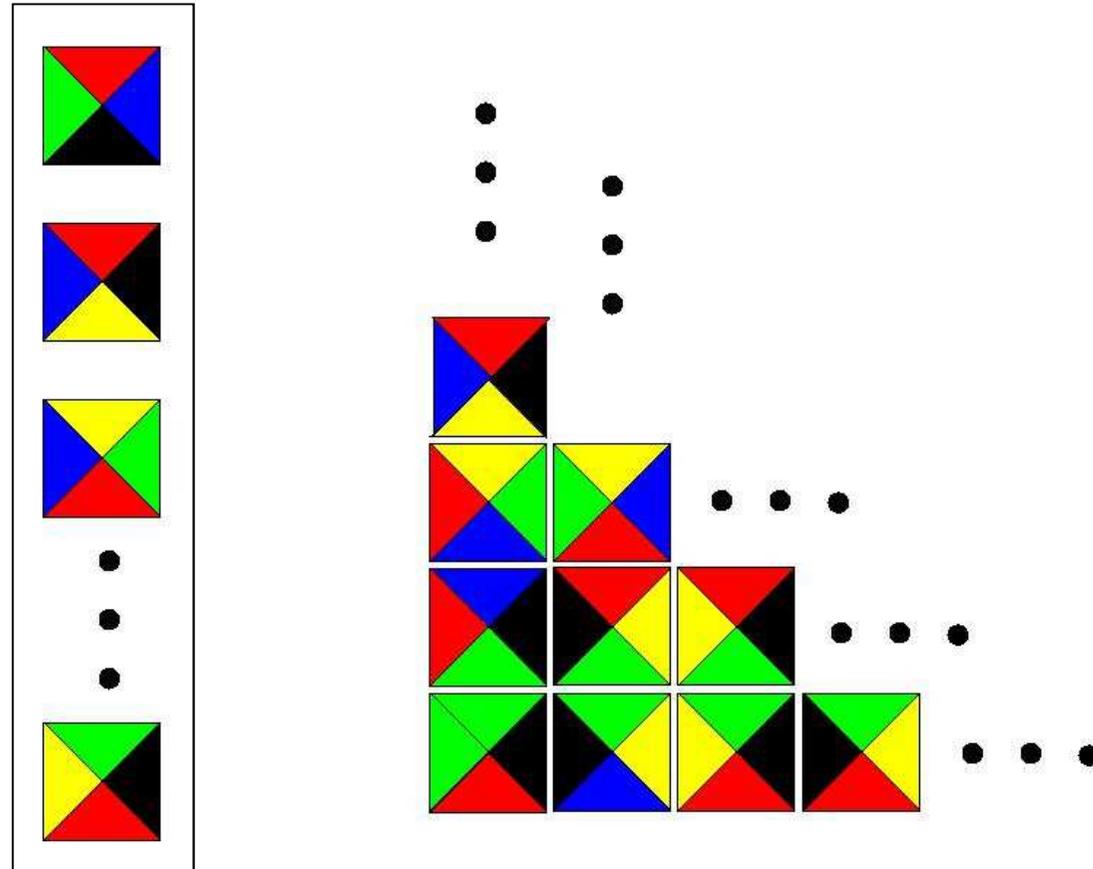
Konjunktive Anfragen mit Ungleichheit

Theorem 6.19

Für gegebene \mathcal{ALC} -Wissensbasis \mathcal{K} und erweiterte konjunktive Anfrage q ist unentscheidbar, ob $\mathcal{K} \models q$.

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 6.20 [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 6.21

Es ist unentscheidbar ob ein gegebenes Dominosystem eine Lösung hat.

Die Reduktion

Ziel:

Gegeben Domino System \mathcal{D} , konstruiere \mathcal{ALC} -Wissensbases $\mathcal{K}_{\mathcal{D}}$ und erweiterte konjunktive Anfrage q , so dass \mathcal{D} Lösung hat gdw $\mathcal{K}_{\mathcal{D}} \not\models q$.

Intuition:

Modelle \mathcal{I} von $\mathcal{K}_{\mathcal{D}}$ so dass *kein* Hom. von q nach $\mathcal{I} \approx$ Lösungen für \mathcal{D} .

“ q -Gegenmodelle”

Signatur von $\mathcal{K}_{\mathcal{D}}$ und q :

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

T6.16

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:

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

3. Benachbarte Kanten haben dieselbe Farbe:

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

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

ABox : leer

T6.17

Die Reduktion

Was noch fehlt:

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

Wir erreichen das mittels q :

$$\begin{aligned} \exists y_0 \exists y_1 \exists y_2 \exists y_3 \exists y'_2 \quad & h(y_0, y_1) \wedge v(y_1, y_2) \wedge \\ & v(y_0, y_3) \wedge h(y_3, y'_2) \wedge y_2 \neq y'_2 \end{aligned} \quad \text{T6.17 cont}$$

Lemma 6.22

\mathcal{D} hat Lösung gdw. $\mathcal{K}_{\mathcal{D}} \not\models q$. T6.18

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

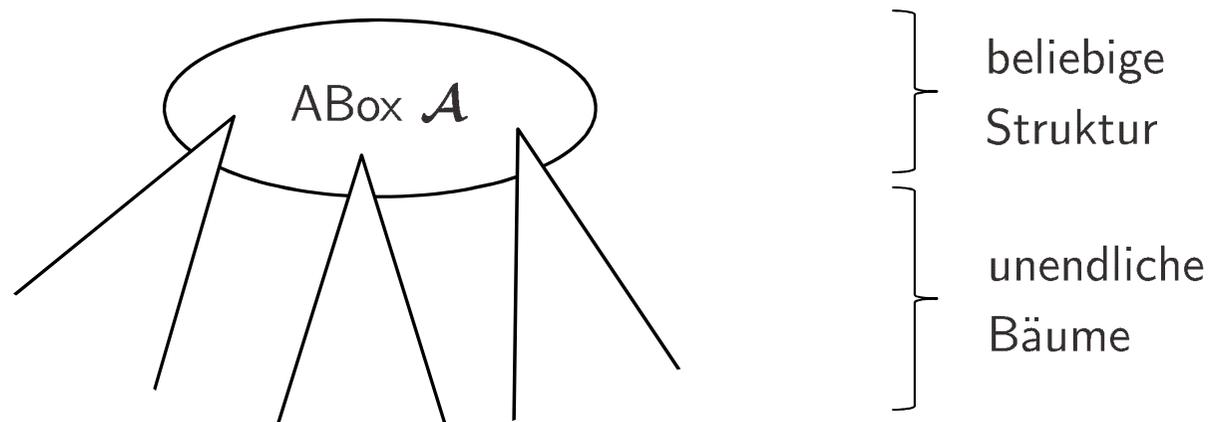
Theorem 6.19 folgt unmittelbar.

Nachbemerkung

Wie konnte so plötzlich aus entscheidbarem Problem unentscheidbares werden?

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

Wenn $\mathcal{K} \not\models q$, dann gibt es immer ein q -Gegenmodell der Form



Die Konstruktion des Gitters zeigt:

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

Zusammenfassung

- In vielen Anwendungen reichen Erfüllbarkeit und Subsumption nicht aus
- ABoxen machen aus DLs “semantische Datenbanken”
- Instanzanfragen sind genauso schwierig wie Erfüllbarkeit
- Konjunktive Anfragen sind ausdrucksstärker, aber in vielen Fällen auch schwerer zu beantworten
- Noch mehr Ausdrucksstärke führt meist zu Unentscheidbarkeit