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

Kapitel 6

ABoxen und Anfragebeantwortung

Ziel des Kapitels

TBoxen repräsentieren allgemeines, begriffliches Wissen

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

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.

Kapitel 6

ABoxen und Anfragebeantwortung

Grundlagen

ABox - Syntax

Von nun an sei

- $\mathbf{N_I}$ eine unendliche Menge von *Individuennamen*

  Diese entsprechen Konstanten im Sinne der Logik erster Stufe

Wir verwendet $a, b, \ldots$ für Individuennamen.

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 $\mathcal{I}$

- bildet jedes $a \in \mathsf{N_I}$ auf Element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ ab;
- *erfüllt* $C(a)$ gdw. $a^{\mathcal{I}} \in C^{\mathcal{I}}$;
- *erfüllt* $r(a, b)$ gdw. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$;

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

T6.1 cont

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)

Kapitel 6

ABoxen und Anfragebeantwortung

Schlussfolgerungsprobleme

Grundlegende Schlussfolgerungsprobleme

Definition 6.3 (Konsistenz, Instanz)
Sei $\mathcal{K}$ Wissensbasis, C Konzept, a Individuenname. 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 $\mathcal{K} \models C(a)$;

Beide Probleme auch ohne (mit leerer) TBox.

Untersch. TBox-Formalismen geben untersch. Entscheidungsprobleme.

Reduktionen

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

Lemma 6.4.

- $\mathcal{K}$ konsistent gdw. $\mathcal{K} \not\models \bot(a)$
- $(\mathcal{T}, \mathcal{A}) \models C(a)$ 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.

Anfragebeantwortung

Viele Anwendungen verwenden ABoxen wie (semantische) Datenbanken

Notation: $\mathsf{Ind}(\mathcal{A})$ ist Menge aller Individuennamen in $\mathcal{A}$

Verschiedene Anfragesprachen möglich:

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

Berechnungsproblem, kein Entscheidungsproblem.

T6.2cont

ABoxen vs Datenbanken

ABoxen repräsentieren *unvollständiges Wissen*:

- Eine ABox hat viele Modelle!
- Es gibt ABoxen $\mathcal{A}$ und Assertionen $C(a)$ so dass

  weder $\mathcal{A} \models C(a)$ noch $\mathcal{A} \models \neg C(a)$

Dies ist die Open World Assumption (OWA) , im Gegensatz zur Closed World Assumption (CWA) bei Datenbanken.

OWA: es sind nur solche Dinge falsch, die explizit als falsch angegeben sind

CWA: was nicht explizit als wahr angegeben ist, ist falsch.

T6.4

ABoxen vs Datenbanken

Anfragebeantwortung

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

Letzteres ist in der Regel wesentlich komplexer

Kapitel 6

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

Vervollständigung

Vervollständigung (Precompletion):

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

Grundideen:

- Zerlege ABox in Teile: ein Teil pro Individuenname $a \in \mathsf{Ind}(\mathcal{A})$.
- Für jedes $a \in \mathsf{Ind}(\mathcal{A})$, 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 $\mathcal{K} = (\mathcal{T}, \mathcal{A})$:

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

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 $a \in \mathsf{Ind}(\mathcal{A}')$. T6.7

Er 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 $\mathcal{ALC}$ also zu erwarten:

- ExpTime-vollständig mit generellen 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

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

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

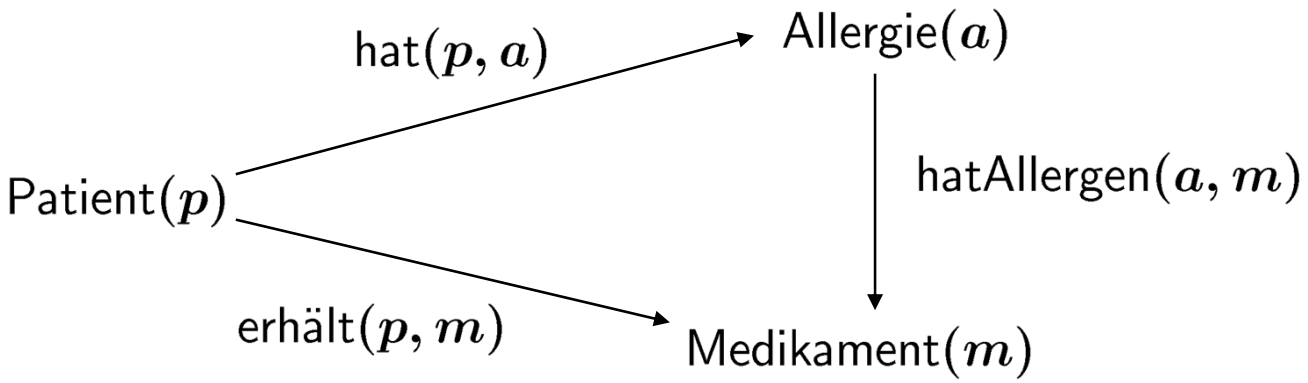
Kapitel 5

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 $C(v)$, mit $v \in \mathbf{N_V}$;
- *Rollenatom* hat die Form $r(v, v')$, mit $v, v' \in \mathbf{N_V}$;

Ein *Atom* ist entweder ein Konzeptatom oder ein Rollenatom

Eine *Konjunktive Anfrage* hat die Form

$$q = \exists y_0, \ldots, y_m \,.\, \alpha_0 \wedge \cdots \wedge \alpha_k$$

wobei $y_0, \ldots, y_m \in \mathbf{N_V}$ und $\alpha_0, \ldots, \alpha_k$ Atome.

Konjunktive Anfrage - Syntax

Sei

$$q = \exists y_0, \ldots, y_m \,.\, \alpha_0 \wedge \cdots \wedge \alpha_k$$

konjunktive Anfrage.

Man unterscheidet zwei Arten von Variablen in q:

- die *quantifizierten Variablen*, also $y_0, \ldots, y_m$
- die *Antwortvariablen*: alle anderen Variablen in $\alpha_0, \ldots, \alpha_k$

T6.11

Konjunktive Anfrage - Notation

Wir schreiben

- $x_0, x_1, \ldots$ für Antwortvariablen
- $y_0, y_1, \ldots$ für quantifizierte Variablen
- φ für Konjunktionen von Atomen
- q, q', etc für konjunktive Anfragen

For konjunktive Anfrage q ist $\mathrm{Var}(q)$ die Menge der Variablen in q.

(Antwortvariablen + quantifizierte Variablen)

Konjunktive Anfrage - Semantik

Definition 6.14. (Treffer, Antwort bzgl. Interpretation)

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

Abildung τ : Var$(q) \to \Delta^{\mathcal{I}}$ ist *Treffer* für q und $\mathcal{I}$ wenn:

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

Antwort auf q bzgl. $\mathcal{I}$: Tupel

$$(a_1, \ldots, a_n) \in \mathsf{N_I}^n$$

so dass es Treffer τ für q und $\mathcal{I}$ 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 $q = \exists \overline{y}.\varphi(\overline{x}, \overline{y})$ mit $\overline{x} = x_1, \ldots, x_n$ und $\mathcal{K}$ Wissensbasis.

Antwort auf q bzgl. $\mathcal{K}$: Tupel

$$(a_1, \ldots, a_n) \in \mathsf{N_I}^n$$

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

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

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

T6.12 cont

Konjunktive Anfragen vs. Instanzanfragen

Konjunktive Anfrage q kann als knoten- und kantenbeschrifteter Graph $(V, E, \mathcal{L})$ gesehen werden:

- $V = \mathsf{Var}(q)$;
- $(v, r, v') \in E$ wenn $r(v, v')$ Atom in q ist;
- $\mathcal{L}(v) = \{C \mid C(v) \text{ ist Atom in } q\}$.

Intuitiv:

Instanzanfragen = baumförmige konjunktive Anfragen.
mit Wurzel einziger Antwortvariable

T6.13

Konjunktive Anfragen und SQL

Datenbanktheorie:

- SQL entspricht FO (wechselseitig polynomiell übersetzbar);
- KAen entsprechen select-project-join queries

In Datenbanksystemen

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

Theorem 6.16.
Die Beantwortung von FO-Anfragen über $\mathcal{ALC}$-Wissensbasen ist unentscheidbar.

T6.14

Boolsche Anfragen

Für Komplexitätsresultate brauchen wir Entscheidungsproblem

Definition 6.17. (Boolsche Anfrage)
Eine konjunktive Anfrage heisst *Boolsch* wenn sie keine Antwortvariablen hat.

T6.15

Boolsche Anfrage liefert keine Individuennamen als Antworten:

ist für jeder Interpretation entweder wahr (= es gibt Treffer)

oder falsch (= es gibt keinen Treffer)

Notation: wir schreiben

- $\mathcal{I} \models q$ wenn q in $\mathcal{I}$ wahr ist;
- $\mathcal{K} \models q$ wenn für jedes Modell $\mathcal{I}$ von $\mathcal{K}$ gilt $\mathcal{I} \models q$
  Wir sagen: q *folgt aus* $\mathcal{K}$

Anfrageimplikation

Anfrageimplikationsproblem:

Gegeben Boolsche KA $\boldsymbol{q}$ und WB $\boldsymbol{\mathcal{K}}$, entscheide ob $\boldsymbol{\mathcal{K}} \models \boldsymbol{q}$.

Anfrageimplikation vs. Anfragebeantwortung:

- Implikation ist Spezialfall von Beantwortung
- Beantwortung $\boldsymbol{n}$-ärer KA kann durch $|\mathsf{Ind}(\boldsymbol{\mathcal{A}})|^n$ Implikationsanfragen realisiert werden

Wir betrachten von nun an Anfrageimplikation statt -beantwortung.

Komplexität und Entscheidbarkeit

Das folgende Resultat ist deutlich schwieriger zu zeigen als die

entsprechenden Resultate für das Instanzproblem.

Theorem 6.18.
Anfrageimplikation ist

- ExpTime-vollständig in $\mathcal{ALC}$ und $\mathcal{ALCQ}$;
- 2ExpTime-vollständig in $\mathcal{ALCI}$ und $\mathcal{ALCQI}$.

(in beiden Fällen mit generellen TBoxen)
Die unteren Schranken gelten bereits, wenn ABox die Form $\{A(a)\}$ hat!

Konjunktive Anfragen mit Ungleichheit

Schon leichte Erweiterungen machen Anfrageimplikation unentscheidbar.

Erweiterte konjunktive Anfrage:

Konjunktive Anfrage mit zusätzlichem Typ von Atom $t \neq t'$.

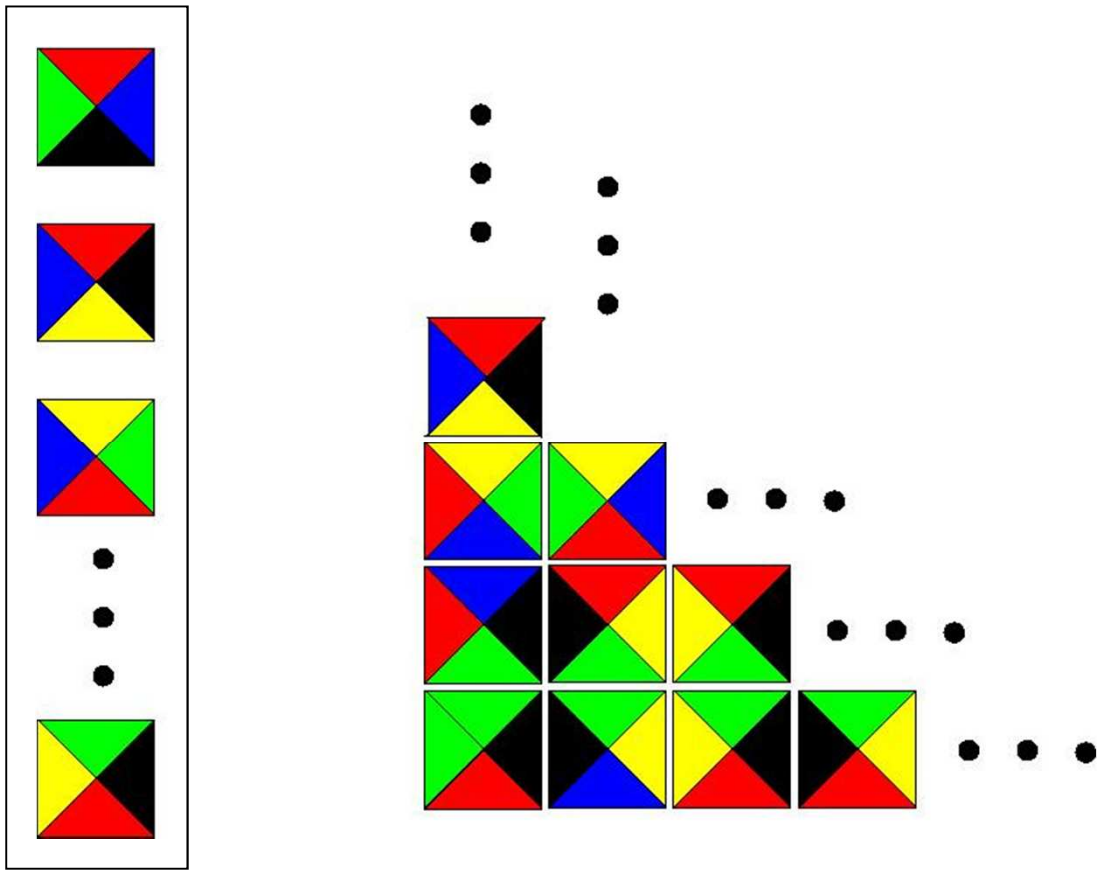
Erweiterte Anfrageimplikation:

Wie Anfrageimplikation, aber mit erweiterten Anfragen. T6.16

Theorem 6.19
Implikation von Boolschen erweiterten konjunktiven Anfragen durch $\mathcal{ALC}$-Wissensbasen ist unentscheidbar.

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 $\pi : \mathbb{N} \times \mathbb{N} \to T$ 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 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 $\mathcal{K}_\mathcal{D}$ mit $\mathcal{I} \not\models q$ repräsentieren Lösungen für $\mathcal{D}$.

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

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

T6.17

Die Reduktion

Jeder Punkt hat horizontalen und vertikalen Nachfolger:

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

Jeder Punkt hat genau einen Typ:

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

Benachbarte Kanten haben dieselbe Farbe:

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

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

Die ABox ist leer!

Die Reduktion

Was noch fehlt:

- die Rollen h und v sind *konfluent*, also:
  jeder hv-Nachfolger ist auch vh-Nachfolger.

Wir erreichen das mittels q:

$$\exists v_0, v_1, v_2, v', v''.h(v_0, v_1), v(v_1, v_2), v(v_0, v'), h(v', v''), v_2 \neq v''$$

T6.18

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

Lemma 6.22
$\mathcal{D}$ hat Lösung gdw. $\mathcal{K}_\mathcal{D} \not\models q$. T6.19

Theorem 6.19 folgt unmittelbar.

Zusammenfassung

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