Logik Teil 2: Prädikatenlogik Grundlagen

Prädikatenlogik

Für viele Zwecke in der Informatik und Mathematik abstrahiert die Aussagenlogik zu stark

Betrachte z.B. die Beispiele aus der Einleitung:

Alle Menschen sind sterblich Sokrates ist ein Mensch	Jedes P ist auch ein Q x is ein P
Sokrates ist sterblich	x ist ein Q

- $\forall n \in \mathbb{N} : \exists n' \in \mathbb{N} : n' = \mathsf{nf}(n)$
- $\forall n \in \mathbb{N} : \mathsf{nf}(n) \neq 0$
- $\cdots$

Bei diesen Aussagen geht es nicht nur um Wahrheitswerte:
Objekte (Menschen, natürliche Zahlen) und Quantifizierung sind zentral!

Prädikatenlogik

Die Prädikatenlogik wurde von Frege gegen Ende des 19Jh eingeführt

Zentrale Elemente:

1. Formeln zusammengesetzt aus Objektvariablen, Boolschen Operatoren und Quantoren

2. eine Semantik, die Objekte und deren Eigenschaften und Beziehungen erfasst

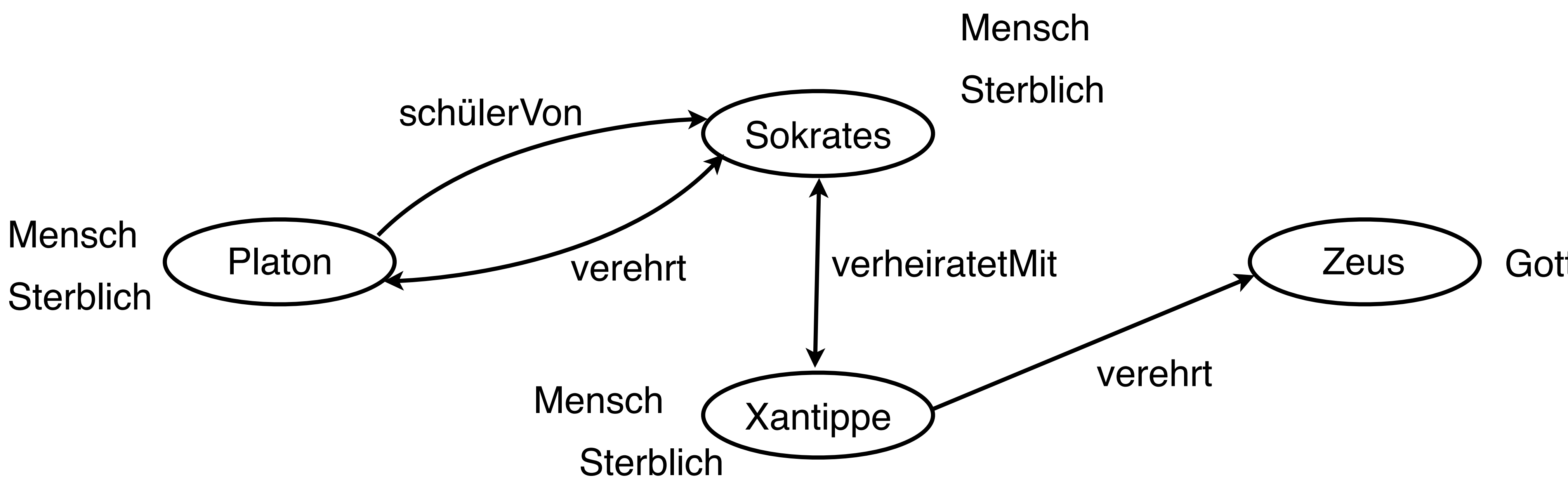
Prädikatenlogik spielt zentrale Rolle in Informatik, Mathematik, Philosophie

Andere Namen: Logik erster Stufe, First-order Logic, Predicate calculus

Abkürzung: FO

Vorschau 1

Eine semantische Struktur der Logik erster Stufe:



Zu dieser Struktur passende Beispielformeln:

$$\forall x\,(\mathsf{Mensch}(x) \to \mathsf{Sterblich}(x))$$

$$\exists x\ \big(\ \exists y\ (\ \mathsf{verehrt}(x,y) \wedge \mathsf{Gott}(y)\ )\ \wedge$$
$$\exists y\ (\ \mathsf{verheiratetMit}(x,y) \wedge \forall z\ (\ \mathsf{vererhrt}(y,z) \to \neg\mathsf{Gott}(z)\ )\ )\ \big)$$

Vorschau 2

Eine semantische Struktur der Logik erster Stufe:

$$0 \xrightarrow{\mathsf{nf}} 1 \xrightarrow{\mathsf{nf}} 2 \xrightarrow{\mathsf{nf}} 3 \xrightarrow{\mathsf{nf}} 4 \dashrightarrow$$

Zu dieser Struktur passende Beispielformeln:

$$\forall x \, \exists y \, (\ y = \mathsf{nf}(x)\ )$$

$$\exists x \, \forall y \, \neg(\ x = \mathsf{nf}(y)\ )$$

$$y = \mathsf{nf}(\mathsf{nf}(x))$$

Übersicht Teil 2

- Kapitel 2.1: Strukturen
- Kapitel 2.2: Syntax und Semantik der Prädikatenlogik
- Kapitel 2.3: Auswertung und Datenbanken
- Kapitel 2.4: Äquivalenz, Erfüllbarkeit, Gültigkeit
- Kapitel 2.4: Prenex-Normalform
- Kapitel 2.6: Unentscheidbarkeit
- Kapitel 2.7: Theorien

Kapitel 2.1: Strukturen

Strukturen

Die Semantik der Prädikatenlogik basiert auf sogenannten Strukturen

Man kann sehr viele Dinge als Struktur repräsentieren:

- Graphen und Hypergraphen
- Wörter (im Sinne der formalen Sprachen)
- Relationale Datenbanken
- Transitionssysteme aus der Hard/Software-Verifikation
- Mathematische Strukturen wie Gruppen, Ringe, Körper
- etc

Dies macht die Prädikatenlogik zu einem sehr generellen Werkzeug

Strukturen

Die Namen, die in einer Struktur verwendet werden, bilden deren Signatur

Definition Signatur

Eine *Signatur* τ ist eine Menge von *Relations*- und *Funktionssymbolen*. Jedes dieser Symbole hat eine feste endliche *Stelligkeit*.

Nullstellige Funktionssymbole nennen wir *Konstantensymbole*.

Beispiel:

- Die Signatur der Arithmetik ist $\{+, \cdot, 0, 1\}$ wobei
  - $+$ und $\cdot$ zweistellige Funktionssymbole
  - 0 und 1 Konstantensymbole

Strukturen

Mehr Beispiele:

- Die Signatur eines gerichteten Graphen ist $\{E\}$, mit E zweistelligem Relationssymbol (das die Kanten repräsentiert)

- Die Signatur einer Datenbank besteht aus je einem n-stelligen Relationssymbol für jede n-spaltige Tabelle

Eine Signatur heisst

- *relational*, wenn sie keine Funktionssymbole enthält
- *funktional*, wenn sie keine Relationssymbole enthält

Die folgenden Definitionen (Struktur, Formel) beziehen sich alle auf eine

Strukturen

Notation: normalerweise verwenden wir:

- P, Q, R für Relationssymbole

  Relationssymbole nennen wir auch *Prädikate*

- f, g, h für Funktionssymbole

- c, d, e für Konstantensymbole

- σ, τ für Signaturen

Statt Stelligkeit sagen wir auch *Arität*

Strukturen

Definition Struktur

Eine *Struktur* $\mathfrak{A}$ besteht aus

- einer nichtleeren Menge A, dem *Universum* von $\mathfrak{A}$
- für jedes n-stellige Relationssymbol P eine Relation $P^{\mathfrak{A}} \subseteq A^n$
- für jedes n-stellige Funktionssymbol f eine Funktion $f^{\mathfrak{A}} : A^n \to A$

Eine Struktur, die genau die Symbole in τ interpretiert, heisst τ*-Struktur*.

Folgendes ist implizit:

- jedes unäre Relationssymbol wird als Teilmenge von A interpretiert
- jedes Funktionssymbol wird als *totale* Funktion interpretiert
- jedes Konstantensymbol wird Element von A interpretiert

Strukturen

Notation:

- Strukturen bezeichnen wir mit Buchstaben in Frakturschrift $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$,
- der entsprechende lateinische Buchstabe A, B, C steht für das Universum der Struktur
- die Elemente des Universums bezeichnen wir mit a, b
- $\mathfrak{A} = (A, P_1^{\mathfrak{A}}, P_2^{\mathfrak{A}}, \ldots, f_1^{\mathfrak{A}}, f_2^{\mathfrak{A}}, \ldots)$ bezeichnet also eine Struktur über der Signatur $\{P_1, P_2, \ldots, f_1, f_2, \ldots\}$ mit Universum A

Strukturen, Graphen, Algebren

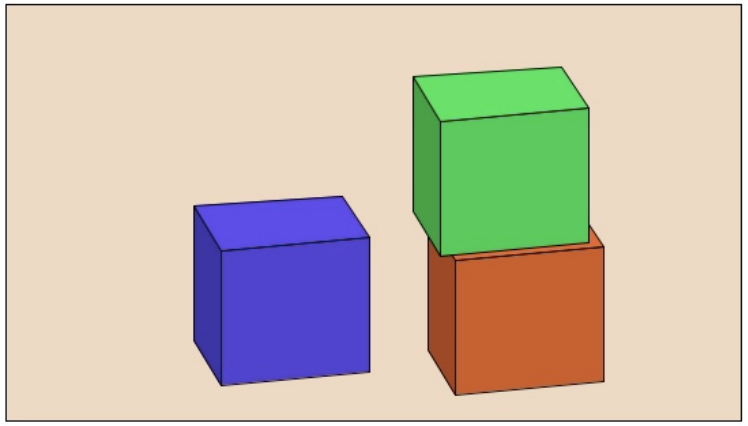
Strukturen generalisieren Graphen und Hypergraphen:

- Struktur $(U, R^{\mathfrak{A}})$ mit R binärem Relationssymbol ist nichts weiter als ein gerichteter Graph (und umgekehrt)
- Strukturen mit mehreren binären Relationssymbolen entsprechen dann kantenbeschrifteten (gerichteten) Graphen
- Unäre Relationssymbole liefern Knotenbeschriftungen im Graph
- n-stellige Relationssymbole mit $n > 2$ entsprechen (gerichteten) Hypergraphen

Strukturen generalisieren zudem Algebren:

Eine Struktur für eine funktionale Signatur ist nichts weiter als eine Algebra (im Sinne der universellen Algebra)

Strukturen - Beispiel 1



repräsentiert als Struktur:

Signatur:

- unäre Relationssymbole Block, R, G, B
- binäre Relationssymbole auf, unter, neben
- Konstantensymbol lieblingsblock

Struktur $\mathfrak{A}$:

- $A = \{rb, gb, bb\}$
- Block$^{\mathfrak{A}} = \{rb, gb, bb\}$, R$^{\mathfrak{A}} = \{rb\}$, G$^{\mathfrak{A}} = \{gb\}$, B$^{\mathfrak{A}} = \{bb\}$
- auf$^{\mathfrak{A}} = \{(gb, rb)\}$, unter$^{\mathfrak{A}} = \{(rb, gb)\}$, neben$^{\mathfrak{A}} = \{(bb, rb), (rb, bb)\}$
- lieblingsblock$^{\mathfrak{A}} = rb$

Strukturen - Beispiel 2

Relationale Datenbank ist eine endliche Sammlung von Tabellen

Jeder Tabelle T zugeordnet ist Spaltenzahl n und Attribute $D_1, \ldots, D_n$

(Attribute z.B. Integers, Strings, etc.)

Konkrete Datenbankinstanz I ordnet dann jedem T endliche Tupelmenge $T^I \subseteq D_1 \times \cdots \times D_n$ zu

I kann als (endliche) Struktur

$$\mathfrak{A}_I = (D, T_1^I, T_2^I, \ldots, T_k^I)$$

repräsentiert werden, wobei

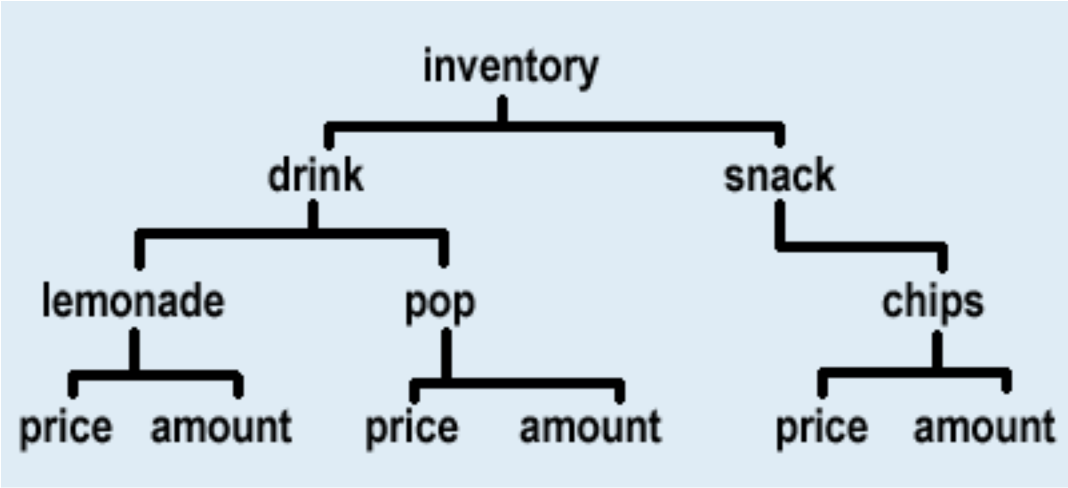
- $T_1, \ldots, T_k$ Relationssymbole für die Tabellen der Datenbank sind
- D die Vereinigung über alle Attribute ist, eingeschränkt auf die (endlich vielen) in I verwendeten Objekte

Strukturen - Beispiel 3

XML-Dokument kann als endliche, baumförmige Struktur gesehen werden

```
<inventory>
    <drink>
        <lemonade>
            <price>$2.50</price>
            <amount>20</amount>
        </lemonade>
        <pop>
            <price>$1.50</price>
            <amount>10</amount>
        </pop>
    </drink>

    <snack>
        <chips>
            <price>$4.50</price>
            <amount>60</amount>
        </chips>
    </snack>
</inventory>
```



Signatur:

binäre Relationssymbole succ für “successor” und

sord für “successor order”

sowie ein unäres Relationssymbole für jedes tag (drink, snack, usw)

Strukturen - Beispiel 4

Strukturen aus der Mathematik, z.B. Arithmetik der natürlichen Zahlen:

$$\mathfrak{N} = (\mathbb{N}, +^{\mathfrak{N}}, \cdot^{\mathfrak{N}}, 0^{\mathfrak{N}}, 1^{\mathfrak{N}}) \qquad \text{(unendlich!)}$$

wobei

- $+^{\mathfrak{N}}, \cdot^{\mathfrak{N}}$ die natürliche Interpretation von $+$ und $\cdot$ sind:

  $$+^{\mathfrak{N}}(x,y) = x + y \qquad \cdot^{\mathfrak{N}}(x,y) = x \cdot y$$

- $0^{\mathfrak{N}} = 0$ und $1^{\mathfrak{N}} = 0$
  (0,1 sowohl Konstantensymbole als auch Elemente des Universums)

Bei offensichtlicher Interpretation lassen wir das $\cdot^{\mathfrak{N}}$ oft weg, also z.B.

$$\mathfrak{N} = (\mathbb{N}, +, \cdot, 0, 1)$$

Analog definiert man z.B. $\mathfrak{R} = (\mathbb{R}, +, \cdot, 0, 1)$ (überabzählbar!)

Strukturen - Beispiel 5

Auch Ordnungen lassen sich als Struktur auffassen, z.B.:

- $\mathfrak{N}_< = (\mathbb{N}, <)$
- $\mathfrak{R}_< = (\mathbb{R}, <)$

(“$<$” binäres Relationssymbol)

In der Informatik werden solche Strukturen oft als Repräsentation von Zeit aufgefasst, die Elemente von $\mathbb{N}$ bzw. $\mathbb{R}$ sind dann die Zeitpunkte

Man kann auch zusätzliche unäre Relationssymbole zulassen, also

$$\mathfrak{A} = (\mathbb{N}, <, P_1^{\mathfrak{A}}, P_2^{\mathfrak{A}}, \dots)$$

wobei eine beliebige Interpretation der $P_1, P_2, \dots$ möglich ist

Mögliche Interpretation:

Jedes P_i repräsentiert eine Aussage (im Sinn der Aussagenlogik), $x \in P_i^{\mathfrak{A}}$ bedeutet ‘”Aussage P_i ist wahr zum Zeitpunkt x”’

Kapitel 2.2: Syntax und Semantik

Syntax

Analog zu den zwei verschiedenen Zutaten von Signaturen und Strukturen (Relationssymbole und Funktionssymbole):

Formeln der Prädikatenlogik bestehen aus zwei Bestandteilen:

- *Terme*, die aus (Objekt)variablen, Konstanten- und Funktionssymbolen gebildet werden
- *Formeln* bestehen dann aus Termen, den Boolschen Operatoren, Quantoren und Relationssymbolen

Wir definieren die Syntax daher in zwei Schritten

Syntax

Wir fixieren eine abzählbar unendliche Menge VAR $= \{x_1, x_2, x_3, \ldots\}$ von *Objektvariablen*.

Definition Term

Die Menge der *Terme* ist induktiv wie folgt definiert:

- jede Variable ist ein Term.
- wenn $t_1, \ldots, t_n$ Terme sind und f ein n-stelliges Funktionssymbol, dann ist auch $f(t_1, \ldots, t_n)$ ein Term

Beachte: jedes Konstantensymbol ist damit ebenfalls ein Term!

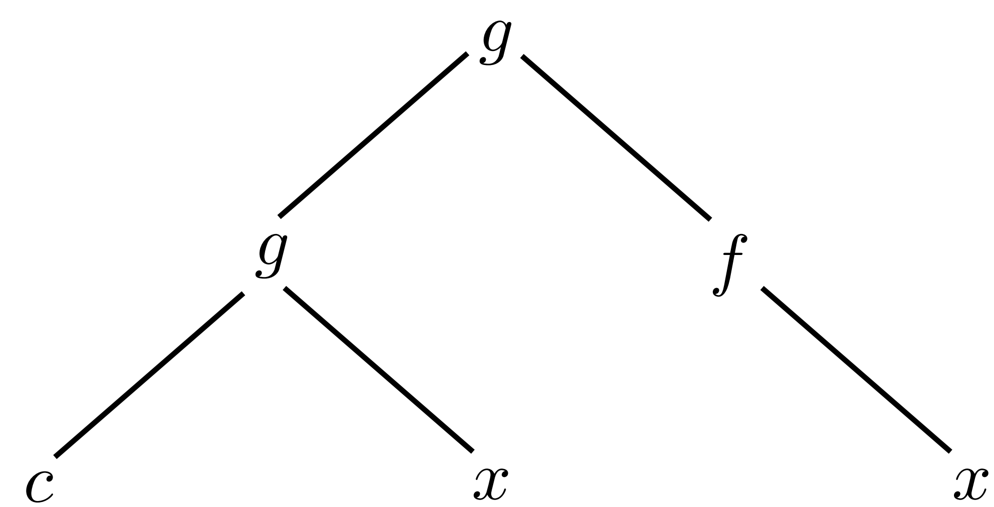
Beispiele:

$$x,\ c,\ f(x),\ g(x,x),\ g(f(x),c),\ g(g(c,c),f(x))$$

$$1 + ((1+1) \cdot 1) \qquad\qquad 1 + ((x+1) \cdot y)$$

Sprechweisen und Konventionen

- Wir bezeichnen Terme mit s und t
- Terme ohne Variablen heissen *Grundterme*, z.B. $1 + ((1 + 1) \cdot 1)$
- Es ist oft nützlich, Terme als Bäume aufzufassen,
  z.B. $g(g(c, x), f(x))$ als



- Für Funktionssymbole wie $+$ und $\cdot$ verwenden wir Infix-Notation,
  also $x + c$ statt $+(x, c)$

Syntax

Definition FO Formeln

Die Menge der *Formeln* der Prädikatenlogik ist induktiv wie folgt definiert:

- sind t_1, t_2 Terme, dann ist $t_1 = t_2$ eine Formel
- sind $t_1, \dots, t_n$ Terme und P ein n-stelliges Relationssymbol, dann ist $P(t_1, \dots, t_n)$ eine Formel
- wenn φ und ψ Formeln sind, dann auch $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$
- wenn φ eine Formel ist und $x \in$ VAR, dann sind $\exists x\, \varphi$ und $\forall x\, \varphi$ Formeln

Die Menge aller Formeln über einer Signatur τ bezeichnen wir mit FO(τ).

Beispiele: $x = c$ $\quad (P(x) \wedge Q(x)) \vee P(y)$ $\quad \forall x\, \exists y\, P(x, f(y))$

$\forall x\ \big(\ \exists y\, \mathsf{neben}(y, x) \vee \exists y\, \mathsf{auf}(y, x)\ \big)$

$\exists x \exists y\ (\ \mathsf{Film}(x, y, \mathsf{Hitchcock}) \wedge \mathsf{Schauspieler}(\mathsf{Connery}, x)\ )$

Sprechweisen und Konventionen

- Atome und deren Negation heissen *Literale*
- Statt $\neg(t = t')$ schreiben wir auch $t \neq t'$
- $\to$ und $\leftrightarrow$ sind analog zur AL definiert
- Klammern werden weggelassen, wenn das Resultat eindeutig ist, wobei $\neg$, $\exists$, $\forall$ stärker binden als $\wedge$ und $\vee$ stärker binden als $\to$, $\leftrightarrow$

  Also z.B. $\exists x\, P(x) \vee Q(x)$ für $(\exists x\, P(x)) \vee Q(x)$,

  nicht für $\exists x\, (P(x) \vee Q(x))$

Freie und gebundene Variablen

Ein *Vorkommen* einer Variable in einer Formel kann durch einen Quantor *gebunden* sein oder nicht (dann ist die Variable *frei*)

Beispiel:

$$\varphi = P(\underset{\text{frei}}{x}) \wedge \exists x\, Q(\underset{\text{gebunden}}{x})$$

Einige Konventionen:

- Wenn wir eine Formel mit $\varphi(x_1, \ldots, x_n)$ bezeichnen, so sind $x_1, \ldots, x_n$ die freien Variablen in φ; für obige Formel: $\varphi(x)$
- Formeln ohne freie Variablen heissen *Satz*

Freie und gebundene Variablen

Formal definiert man die Menge der freien Variablen wie folgt.

Mit $\mathsf{Var}(\varphi)$ bezeichnen wir die Menge der in der Formel φ vorkommenden Variablen.

Definition Freie Variable

Sei φ eine Formel. Die Menge $\mathsf{Frei}(\varphi)$ der *freien Variablen* von φ ist induktiv wie folgt definiert:

- Für atomare Formeln φ ist $\mathsf{Frei}(\varphi) = \mathsf{Var}(\varphi)$
- $\mathsf{Frei}(\neg\varphi) = \mathsf{Frei}(\varphi)$
- $\mathsf{Frei}(\varphi \wedge \psi) = \mathsf{Frei}(\varphi \vee \psi) = \mathsf{Frei}(\varphi) \cup \mathsf{Frei}(\psi)$
- $\mathsf{Frei}(\exists x\, \varphi) = \mathsf{Frei}(\forall x\, \varphi) = \mathsf{Frei}(\varphi) \setminus \{x\}$

Semantik

Stuktur interpretiert nur Symbole, aber keine Variablen; dafür Zuweisung

Definition Zuweisung

Sei $\mathfrak{A}$ eine Struktur. Eine *Zuweisung in* $\mathfrak{A}$ ist eine Abbildung β : VAR $\rightarrow A$.

Man erweitert β wie folgt induktiv auf τ-Terme:

- wenn $t = f(t_1, \ldots, t_k)$, dann $\beta(t) = f^{\mathfrak{A}}(\beta(t_1), \ldots, \beta(t_k))$

Ein Paar $(\mathfrak{A}, \beta)$ mit β Zuweisung in $\mathfrak{A}$ heisst *Interpretation*.

Beachte: der implizite Induktionsanfang ist:

- wenn $t = x \in$ VAR, dann $\beta(t) = \beta(x)$
- wenn $t = c \in F^0$, dann $\beta(t) = c^{\mathfrak{A}}$

Semantik

Definition Semantik von FO

Wir definieren Erfülltheitsrelation $\models$ zwischen Interpretationen $(\mathfrak{A}, \beta)$ und FO-Formeln induktiv wie folgt:

- $\mathfrak{A}, \beta \models t = t'$ gdw. $\beta(t) = \beta(t')$
- $\mathfrak{A}, \beta \models P(t_1, \dots, t_n)$ gdw. $(\beta(t_1), \dots, \beta(t_n)) \in P^{\mathfrak{A}}$
- $\mathfrak{A}, \beta \models \neg\varphi$ gdw. $\mathfrak{A}, \beta \not\models \varphi$
- $\mathfrak{A}, \beta \models \varphi \wedge \psi$ gdw. $\mathfrak{A}, \beta \models \varphi$ und $\mathfrak{A}, \beta \models \psi$
- $\mathfrak{A}, \beta \models \varphi \vee \psi$ gdw. $\mathfrak{A}, \beta \models \varphi$ oder $\mathfrak{A}, \beta \models \psi$
- $\mathfrak{A}, \beta \models \exists x\, \varphi$ gdw. ein $a \in A$ existiert mit $\mathfrak{A}, \overbrace{\beta[x/a]}^{\text{Wie } \beta\text{, außer } x \mapsto a} \models \varphi$
- $\mathfrak{A}, \beta \models \forall x\, \varphi$ gdw. für alle $a \in A$ gilt, dass $\mathfrak{A}, \beta[x/a] \models \varphi$

Wenn $\mathfrak{A}, \beta \models \varphi$, dann ist $(\mathfrak{A}, \beta)$ ein *Modell* für φ.

Koinzidenzlemma

Analog zur Aussagenlogik: $(\mathfrak{A}, \beta) \models \varphi$ ist unabhängig von Interpretation der Symbolen und Variablen, die in φ gar nicht (bzw nicht frei) vorkommen.

$\mathsf{sig}(\varphi)$ bezeichne die Signatur der Formel φ, also die Menge der in φ vorkommenden Relations- und Funktionssymbole

Koinzidenzlemma

Sei φ eine FO Formel und $(\mathfrak{A}, \beta)$, $(\mathfrak{A}', \beta')$ Interpretationen so dass

- $A = A'$;
- $S^{\mathfrak{A}} = S^{\mathfrak{A}'}$ für alle $S \in \mathsf{sig}(\varphi)$
- für alle $x \in \mathsf{Frei}(\varphi)$ gilt: $\beta(x) = \beta'(x)$

Dann $\mathfrak{A}, \beta \models \varphi$ gdw. $\mathfrak{A}', \beta' \models \varphi$

Beweis per Induktion über die Struktur von φ.

Koinzidenzlemma

Wenn wir mit einer Formel φ arbeiten, so erlaubt uns das Koinzidenz-lemma, in Zuweisungen nur die Variablen Frei(φ) zu betrachten.

Das erlaubt insbesondere folgende Notation:

Für eine Formel $\varphi(x_1, \ldots, x_k)$ schreiben wir

$$\mathfrak{A} \models \varphi[a_1, \ldots, a_k]$$

wenn $\mathfrak{A}, \beta \models \varphi$, wobei $\beta(x_i) = a_i$ für $1 \leq i \leq k$

Wenn φ Satz ist, dann wird daraus einfach $\mathfrak{A} \models \varphi$

Isomorphielemma

Es existiert ein Isomorphismus zwischen zwei Stukturen gdw. diese sich nur durch Umbennen der Elemente des Universums unterscheiden

Definition Isomorphismus

Seien $\mathfrak{A}$ und $\mathfrak{B}$ Strukturen. Bijektion $\pi : A \to B$ ist *Isomorphismus*, wenn folgende Bedingungen erfüllt sind:

- Für jedes n-stellige Relationssymbol P und alle $a_1, \dots, a_n \in A^n$:

$$(a_1, \dots, a_n) \in P^{\mathfrak{A}} \text{ gdw. } (\pi(a_1), \dots, \pi(a_n)) \in P^{\mathfrak{B}}$$

- Für jedes n-stellige Funktionssymbol f und alle $a_1, \dots, a_n \in A^n$:

$$\pi(f^{\mathfrak{A}}(a_1, \dots, a_n)) = f^{\mathfrak{B}}(\pi(a_1), \dots, \pi(a_n))$$

Beispiel

Isomorphielemma

Isomorphielemma

Seien $\mathfrak{A}$, $\mathfrak{B}$ Strukturen und $\pi : A \to B$ ein Isomorphismus.

Dann gilt für all Formeln $\varphi(x_1, \ldots, x_n)$ und alle $a_1, \ldots, a_n \in A$:

$$\mathfrak{A} \models \varphi[a_1, \ldots, a_n] \text{ gdw. } \mathfrak{B} \models \varphi[\pi(a_1), \ldots, \pi(a_n)]$$

Insbesondere gilt also für alle Sätze φ: $\mathfrak{A} \models \varphi$ gdw. $\mathfrak{B} \models \varphi$.

Intuitiv:

- FO kann nicht zwischen isomorphen Strukturen unterscheiden
- Die Namen der Elemente des Universums sind Schall und Rauch

Prädikatenlogik

Kapitel 2.3: Auswertung und Datenbanken

Auswertung

Definition Auswertungsproblem

Das *Auswertungsproblem der Prädikatenlogik* ist:

Gegeben: FO Formel $\varphi(x_1, \dots, x_n)$, endliche Interpretation $(\mathfrak{A}, \beta)$
mit β Zuweisung für $x_1, \dots, x_n$

Frage: Gilt $\mathfrak{A}, \beta \models \varphi$?

Theorem

Das Auswertungsproblem der Prädikatenlogik erster Stufe ist PSpace-vollständig.

Wir wollen hier nur Entscheidbarkeit in PSpace beweisen

PSpace-Härte zeigt man über eine Reduktion von QBF,

siehe VL Komplexitätstheorie

Auswertung

ausw$(\mathfrak{A}, \beta, \varphi)$

case

$\varphi = (t = t')$: return 1 if $\beta(t) = \beta(t')$, else return 0

$\varphi = P(t_1, \dots, t_k)$: return 1 if $(\beta(t_1), \dots, \beta(t_k)) \in P^{\mathfrak{A}}$, else return 0

$\varphi = \neg\psi$: return $1-$ausw$(\mathfrak{A}, \beta, \psi)$

$\varphi = \psi \wedge \vartheta$: return $\min\{$ausw$(\mathfrak{A}, \beta, \psi),$ ausw$(\mathfrak{A}, \beta, \vartheta)\}$

$\varphi = \psi \vee \vartheta$: return $\max\{$ausw$(\mathfrak{A}, \beta, \psi),$ ausw$(\mathfrak{A}, \beta, \vartheta)\}$

$\varphi = \exists x\, \psi$:

rufe ausw$(\mathfrak{A}, \beta[x/a], \psi)$ für alle $a \in A$

return 1 if ein Ruf erfolgreich, else return 0

$\varphi = \forall x\, \psi$:

rufe ausw$(\mathfrak{A}, \beta[x/a], \psi)$ für alle $a \in A$

return 1 if alle Rufe erfolgreich, else return 0

endcase

Auswertung

Lemma

Der Algorithmus

1. ist korrekt: ausw$(\mathfrak{A}, \beta, \varphi) = 1$ gdw. $\mathfrak{A}, \beta \models \varphi$
2. benötigt nur polynomiell viel Platz

Für den Beweis:

Die *Schachtelungstiefe* $st(\varphi)$ einer Formel φ is induktiv definiert wie folgt:

- $st(t = t') = st(P(t_1, \ldots, t_k)) = 0$
- $st(\neg\varphi) = st(\exists x\, \varphi) = st(\forall x\, \varphi) = st(\varphi) + 1$
- $st(\varphi \wedge \psi) = st(\varphi \vee \psi) = \max\{st(\varphi), st(\psi)\} + 1$

PS: Der Algorithmus benötigt natürlich exponentiell viel Zeit

FO und Datenbanken

Man kann FO auf natürliche Weise als Anfragesprache für DBen sehen:

- schon gesehen: Datenbankinstanz $\approx$ relationale Struktur
- Antwort auf FO-Anfrage $\varphi(x_1, \ldots, x_n)$ bzgl. Datenbankinstanz $\mathfrak{A}$:

$$\mathsf{ans}(\mathfrak{A}, \varphi) = \{(a_1, \ldots, a_n) \in A^n \mid \mathfrak{A} \models \varphi[a_1, \ldots, a_n]\}$$

Film:

Titel	Jahr	Regisseur
Die Vögel	1963	Hitchcock
Marnie	1964	Hitchcock
Goldfinger	1964	Hamilton

Schauspieler:

Name	Titel
Connery	Marnie
Connery	Goldfinger
Hedren	Die Vögel

$\varphi = \exists y\ (\ \mathsf{Film}(\underline{x}, y, \mathsf{Hitchcock}) \wedge \mathsf{Schauspieler}(\mathsf{Connery}, \underline{x})\ )$

$\mathsf{ans}(\mathfrak{A}, \varphi) = \{\mathsf{Marnie}\}$

FO und Datenbanken

In diesem Zusammenhang wird FO auch das *relationale Kalkül* genannt

FO/das relationale Kalkül ist im wesentlichen nichts anderes als SQL!!

Beispiele:

SELECT Titel FROM Film WHERE Regisseur = Hitchcock

$\exists y\, \text{Film}(\underline{x}, y, \text{Hitchcock})$

SELECT Name,Jahr FROM Schauspieler,Film
WHERE Schauspieler.Titel = Film.Titel

$\exists z \exists z' \; (\; \text{Schauspieler}(\underline{x}, z) \wedge \text{Film}(z, \underline{y}, z') \;)$

FO und Datenbanken

Sei *Kern-SQL* die Einschränkung von SQL auf

SELECT FROM WHERE (in Bedingungen sind = und AND erlaubt),
UNION,
MINUS

Die meisten anderen Elemente von SQL dienen nur der Benutzbarkeit, erhöhen aber nicht die Ausdrucksstärke

Nicht sehr schwer: jede Kern-SQL Anfrage kann in äquivalente FO-Anfrage übersetzt werden (äquivalent = dieselben Antworten auf jeder Datenbank)

Für die Übersetzung FO $\Rightarrow$ SQL muß man eine Einschränkung machen:

Domänenunabhängigkeit der FO-Anfrage

FO und Datenbanken

Intuitiv: Anfrage ist domänenunabhängig wenn Antworten nicht von Elementen abhängen, die in *keiner* Relation/Tabelle vorkommen

Definition Domänenunabhängigkeit

Eine FO-Formel φ ist *domänenunabhängig* wenn für alle Strukturen $\mathfrak{A} = (A, P_1^{\mathfrak{A}}, P_2^{\mathfrak{A}}, \dots)$ und $B \supseteq A$ gilt:

$$\mathsf{ans}(\mathfrak{A}, \varphi) = \mathsf{ans}((B, P_1^{\mathfrak{A}}, P_2^{\mathfrak{A}}, \dots), \varphi).$$

Domänenabhängige FO-Anfragen sind meist sinnlos:

$$\neg \exists y \, \mathsf{Schauspieler}(\underline{x}, y)$$

liefert alle in der Datenbank verwendeten Strings
und Zahlen ausser Connery und Hedren

FO und Datenbanken

Folgendes Resultat von 1970 ist die Grundlage für die Entwicklung der Relationalen Datenbanksysteme

(Codd arbeitete bei IBM, implementierte die erste relationale Datenbank “System R”)

Theorem (Codd)

Jede domänenunabhängige FO-Anfrage ist äquivalent zu einer Anfrage in Kern-SQL und umgekehrt. Die Übersetzung benötigt nur lineare Zeit.

Unser Algorithmus für FO-Auswertung kann also auch zur SQL-Anfrage-beantwortung verwendet werden!

Kapitel 2.4: Äquivalenz, Erfüllbarkeit, Gültigkeit

Äquivalenz

Definition Äquivalenz

Zwei FO Formeln φ and ψ mit $\mathsf{Frei}(\varphi) = \mathsf{Frei}(\psi)$ sind *äquivalent* wenn für alle Interpretationen $(\mathfrak{A}, \beta)$ gilt, dass $\mathfrak{A}, \beta \models \varphi$ gdw. $\mathfrak{A}, \beta \models \psi$.
Wir schreiben dann $\varphi \equiv \psi$.

Der Begriff einer *Teilformel* einer FO Formel kann auf die offensichtliche Weise induktiv definiert werden, analog zur Aussagenlogik.

Auch in FO sind äquivalente Formeln austauschbar:

Ersetzungslemma

Seien φ and ψ äquivalente FO Formeln, ϑ eine Formel mit $\varphi \in \mathsf{TF}(\vartheta)$ und ϑ' eine Formel, die sich aus ϑ ergibt, indem ein beliebiges Vorkommen von φ durch ψ ersetzt wird. Dann gilt $\vartheta \equiv \vartheta'$.

Beweis per Induktion über die Struktur von ϑ.

Äquivalenz

Leicht zu sehen: alle Äquivalenzen aus der Aussagenlogik gelten auch in FO, z.B.:

$\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$ für beliebige FO-Formeln φ, ψ

Natürlich gibt es auch interessante FO-spezifische Äquivalenzen, z.B.

- $\forall x\, \varphi \equiv \neg\exists x\, \neg\varphi$ (Dualität von $\exists$ und $\forall$)
- $\exists x\, (\varphi \vee \psi) \equiv \exists x\, \varphi \vee \exists x \psi$ ($\exists$ distribuiert über $\vee$)
- $\forall x\, (\varphi \wedge \psi) \equiv \forall x\, \varphi \wedge \forall x \psi$ ($\forall$ distribuiert über $\wedge$)
- $\exists x\, \exists y \varphi \equiv \exists y\, \exists x\, \varphi$
- $\forall x\, \forall y \varphi \equiv \forall y\, \forall x\, \varphi$

Äquivalenz

FO-Formel heisst *reduziert*, wenn sie nur die Junktoren $\neg$, $\wedge$ und nur den Quantor $\exists$ enthält

Lemma

Jede FO-Formel kann in Linearzeit in eine äquivalente *reduzierte* FO-Formel gewandelt werden.

Beweis klar wegen

$$\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$$

$$\forall x\, \varphi \equiv \neg\exists x\, \neg\varphi$$

In Induktionsbeweisen müssen wir also nur $\neg, \wedge, \exists$ betrachten

Erfüllbarkeit, Gültigkeit, Konsequenz

Folgende Begriffe sind exakt analog zur Aussagenlogik:

Definition Erfüllbarkeit, Gültigkeit, Konsequenz

Ein Satz φ heißt

- *erfüllbar*, wenn er ein Modell hat (Struktur, die ihn wahr macht)
- *gültig* oder *Tautologie*, wenn jede Struktur ein Modell von φ ist
- *Konsequenz* von Satz ψ wenn für alle Strukturen $\mathfrak{A}$ mit $\mathfrak{A} \models \psi$ auch $\mathfrak{A} \models \varphi$ gilt (wir schreiben dann $\psi \models \varphi$)

Man kann diese Begriffe natürlich auch für Formeln mit freien Variablen definieren, verwendet dann Interpretationen statt Strukturen

Kapitel 2.5: Prenex-Normalform

Pränex-Normalform

FO Formel φ ist *bereinigt* wenn

- keine Variable in φ sowohl frei als auch gebunden auftritt
- keine Variable mehr als einmal quantifiziert wird

Jede Formel kann leicht durch *Umbenennung quantifizierter Variablen* bereinigt werden, z.B.:

$$\exists y\,(P(\underline{x},y) \land \forall x\, Q(x,y)) \quad \text{äquivalent zu} \quad \exists y\,(P(\underline{x},y) \land \forall z\, Q(z,y))$$

Definition Pränex-Normalform

Eine FO Formel φ ist in *Pränex-Normalform (PNF)* wenn sie bereinigt ist und und die Form

$$Q_1 x_1 \cdots Q_n x_n\, \varphi$$

hat wobei $Q_i \in \{\exists, \forall\}$ und φ quantorenfrei.

Pränex-Normalform

Theorem

Jede FO Formel kann in Linearzeit in eine äquivalente Formel in PNF gewandelt werden.

Für den Beweis benötigen wir folgende Äquivalenzen:

Falls x nicht frei in φ vorkommt, gilt:

- $\varphi \lor \exists x\, \psi \equiv \exists x\, (\varphi \lor \psi)$
- $\varphi \land \exists x\, \psi \equiv \exists x\, (\varphi \land \psi)$
- $\varphi \lor \forall x\, \psi \equiv \forall x\, (\varphi \lor \psi)$
- $\varphi \land \forall x\, \psi \equiv \forall x\, (\varphi \land \psi)$

Beispiel

Kapitel 2.6: Unentscheidbarkeit

Unentscheidbarkeit

Bis in die 1930er hofften viele Mathematiker, dass die Prädikatenlogik oder ähnlich ausdrucksstarke Logiken entscheidbar sein würden.

Besonders prominent ist Hilbert, der 1928 die Lösung des "Entscheidungsproblems" der Logik als eines der wichtigsten offenen Probleme der Mathematik bezeichnet hat.

Da wichtige Teile der Mathematik in FO formalisierbar (z.B. Gruppentheorie): viele manuelle mathematische Beweise könnten durch automatische ersetzt werden.

Aber das wäre dann doch zu schön, um wahr zu sein. :)

Lesetipp


LOGICOMIX
EINE EPISCHE SUCHE NACH WAHRHEIT
APOSTOLOS DOXIADIS UND CHRISTOS H. PAPADIMITRIOU
GRAFIK VON ALECOS PAPADATOS UND ANNIE DI DONNA
ÜBERSETZT VON EBI NAUMANN
ATRIUM

Unentscheidbarkeit

Wir zeigen, dass die Gültigkeit von FO-Formeln unentscheidbar ist.

Unentscheidbarkeit von Erfüllbarkeit und Konsequenz folgt dann per Reduktion von Gültigkeit, denn analog zur Aussagenlogik gilt:

- Satz φ ist gültig gdw. $\neg\varphi$ unerfüllbar ist
- Satz φ ist gültig gdw. $\varphi_{\mathsf{taut}} \models \varphi$ wobei φ_{taut} beliebige Tautologie

Der Beweis ist per Reduktion des Post'schen Korrespondenzproblems

Unentscheidbarkeit

Wir verwenden eine Reduktion des Postschen Korrespondenzproblems

Definition Postsches Korrespondenzproblem (PCP)

Gegeben: Eine Folge $F = (u_1, v_1), \ldots, (u_k, v_k)$ von Wortpaaren, mit $u_i, v_i \in \{0, 1\}^*$

Frage: Gibt es eine Indexfolge $i_1, \ldots, i_\ell$ so dass $u_{i_1} \cdots u_{i_\ell} = v_{i_1} \cdots v_{i_\ell}$?

Eine solche Folge heisst *Lösung* für F.

Bekannt aus VL “Theoretische Informatik 2”:

Theorem (Post)

Das PCP ist unentscheidbar.

Unentscheidbarkeit

Ziel: Für gegebenes PCP F einen FO-Satz φ_F konstruieren, so dass

F hat eine Lösung gdw. φ_F gültig ist.

Verwendete Signatur:

- ein Konstantensymbol c_ε
- zwei einstellige Funktionssymbole f_0 und f_1
- ein zweistelliges Relationssymbol P

Intuition:

- c_ε, f_0, f_1 erzeugen alle Wörter
- P kennzeichnet Wortpaare, die F erzeugen kann

Unentscheidbarkeit

Schreibweise:

für Wort $w = w_1 \cdots w_n \in \{0,1\}^*$ steht $t_w(x)$ für $f_{w_n}(f_{w_{n-1}}(\cdots f_{w_1}(x)))$

Für PCP $F = (u_1, v_1), \ldots, (u_k, v_k)$ setze

$$\varphi_F \quad = \quad (\varphi \wedge \psi) \rightarrow \exists x \, P(x,x)$$

wobei

$$\varphi \quad = \quad \bigwedge_{i=1..k} P(t_{u_i}(c_\varepsilon), t_{v_i}(c_\varepsilon))$$

$$\psi \quad = \quad \forall x \, \forall y \, (P(x,y) \rightarrow \bigwedge_{i=1..k} P(t_{u_i}(x), t_{v_i}(y))$$

Lemma

F hat eine Lösung gdw. φ_F gültig ist.

Unentscheidbarkeit

Theorem (Church,Turing)

In FO sind Gültigkeit, Erfüllbarkeit, Konsequenz unentscheidbar.

Unentscheidbarkeit gilt auch für *relationale* Signaturen:

- ersetze c_ε durch unäres Prädikat A_ε
- ersetze f_0, f_1 durch binäre Prädikate P_0, P_1
- erzwinge das “richtige Verhalten”:

$$\exists x \, \big(\, A_\varepsilon(x) \wedge \forall y \, (A_\varepsilon(y) \rightarrow x = y) \, \big)$$

$$\forall x \, \exists y \, P_i(x, y)$$

$$\forall x \forall y \forall z \, ((P_i(x, y) \wedge P_i(x, z)) \rightarrow y = z)$$

für $i \in \{1, 2\}$

Beachte: Datenbanken haben rein relationale Signaturen.

Auch die Gleichheit ist durch `normales Relationssymbol’ simulierbar

Unentscheidbarkeit

Beachte:
die vorgestellte Reduktion erfordert unendliche Modelle, in der Informatik benötigt man aber meist nur endliche Modelle (z.B. Datenbanken)

Das liefert unterschiedliche Begriffe von Erfüllbarkeit, Tautologie, etc

Z.B. ist folgende Formel erfüllbar, aber nicht endlich erfüllbar

$$\begin{aligned}
&\forall x\, \neg R(x, c) \wedge \\
&\forall x\, \exists y\, R(x, y) \wedge \\
&\forall x \forall x' \forall y\, (\, R(x, y) \wedge R(x', y) \to x = x'\,)
\end{aligned}$$

Ihre Negation ist also eine Tautologie in endlichen Modellen,
aber nicht im allgemeinen.

Unentscheidbarkeit

Theorem (Trakhtenbrot)

Folgende Probleme sind unentscheidbar:

- Endliche Gültigkeit:
  Ist eine FO-Formel in allen endlichen Interpretationen erfüllt?
- Endliche Erfüllbarkeit:
  Hat eine FO-Formel ein endliches Modell?
- Endliche Konsequenz:
  Gilt für zwei FO-Formeln φ, ψ und alle endlichen Interpretationen $(\mathfrak{A}, \beta)$ mit $\mathfrak{A}, \beta \models \varphi$ auch $\mathfrak{A}, \beta \models \psi$?

Beweis zum Beispiel durch Reduktion des Halteproblems

Unentscheidbarkeit

Da alle Formeln im Beweis von Trakhtenbrots Theorem domänenunabhängig sind, sind auch folgende SQL-Probleme unentscheidbar:

- Gegeben eine SQL-Anfrage, entscheide ob es eine Datenbank-Instanz gibt, für die die Anfrage eine nicht-leere Antwort liefert

- Gegeben zwei SQL-Anfragen, entscheide ob für jede Datenbank-Instanz gilt: die Antwort für die erste Anfrage ist eine Teilmenge der Antwort für die zweite Anfrage (*Query containment*)

- Gegeben zwei SQL-Anfragen, entscheide ob sie für alle Datenbankinstanzen dieselben Antworten liefern.

Diese Probleme sind von praktischer Bedeutung z.B. für die Anfrageoptimierung in relationalen Datenbanksystemen

Unentscheidbarkeit

Es gibt aber auch positives zu berichten:

- Die gültigen FO-Formeln sind *rekursiv aufzählbar* (Teil 3), dies ist die Grundlage für automatisches Theorembeweisen

  Intuitiv:

  - Wenn ich den Beweiser nach einem wahren mathematischen Theorem frage, findet er schließlich einen Beweis
  - Wenn ich den Beweiser nach einem nicht gültigen Theorem frage, terminiert er nicht

- Über verschiedenen wichtigen Strukturklassen wie Wörtern und Bäumen (und sogar die Logik 2. Stufe) erhält man Entscheidbarkeit (Teil 4)

  Wichtige Anwendungen in der Verifikation

Unentscheidbarkeit

Es gibt aber auch positives zu berichten:

- Verschiedene syntaktische Einschränkungen liefern Entscheidbarkeit

    1. Nur unäre Relationssymbole, keine Funktionssymbole
    2. Nur 2 Variablen statt unendlich viele
    3. Formeln in PNF mit eingeschränktem Quantorenpräfix, z.B. $\exists^*\forall^*$
    4. Guarded Fragment: bei $\exists x\, \varphi$ und $\forall x\, \varphi$ Form von φ eingeschränkt

  Wichtige Anwendungen in der KI und der Verifikation

- Im folgenden: verschiedene wichtige FO-Theorien sind entscheidbar

  Wichtig z.B. für das Theorembeweisen in der Mathematik

Kapitel 2.7: Theorien

FO Theorien

Manche Strukturen haben eine besonders große Bedeutung, z.B.:

Arithmetik der natürlichen Zahlen $(\mathbb{N}, +, *, 0, 1)$

Die in dieser Struktur erfüllten FO-Sätze sind mathematisch von grosser Bedeutung.

Bereits gesehen:
die Existenz unendlich vieler Primzahl-Zwillinge ist in FO beschreibbar

Weiteres Beispiel: Goldbachs Vermutung

$$\forall x \, (x > 2 \land \mathsf{Even}(x) \rightarrow \exists y \exists y' \, (\mathsf{Prim}(y) \land \mathsf{Prim}(y') \land x = y + y'))$$

wobei $\mathsf{Even}(x) = \exists y \, (x = y + y)$, etc.

Theorien sind “kohärente” Mengen von FO-Sätzen, z.B. diejenigen, die in einer ausgewählten Struktur wahr sind.

FO Theorien

Mengen von Formeln (endlich oder unendlich):
Begriffe wie Modell, Erfüllbarkeit und Konsequenz übertragen sich leicht

Z.B. ist $\mathfrak{A}$ *Modell* für Menge Γ von Sätzen gdw. $\mathfrak{A} \models \varphi$ für alle $\varphi \in \Gamma$

Für die folgende Definition nehmen wir (implizit) eine feste Signatur τ an, über der alle Formeln gebildet werden.

Definition FO Theorie

Eine *FO-Theorie* ist eine erfüllbare Menge T von FO-Sätzen, die unter Konsequenz abgeschlossen sind:

$$T \models \varphi \text{ impliziert } \varphi \in T \text{ für alle Sätze } \varphi$$

T heißt *vollständig* wenn für alle Sätze φ gilt: $\varphi \in T$ oder $\neg\varphi \in T$

FO Theorien

Beispiele:

1. Menge aller Tautologien (in einer fixen Signatur τ) ist FO-Theorie

   enthalten in allen anderen Theorien, nicht vollständig

2. Sei $\mathfrak{A}$ eine τ-Stuktur. Dann ist

$$\mathsf{Th}(\mathfrak{A}) = \{\varphi \text{ ist } \tau\text{-Satz} \mid \mathfrak{A} \models \varphi\}$$

   eine vollständige FO-Theorie.

3. Wenn Ω erfüllbare Menge von FO-Sätzen, dann ist

$$\mathsf{Abschluss}(\Omega) = \{\varphi \text{ ist } \tau\text{-Satz} \mid \Omega \models \varphi\}$$

   FO-Theorie (im allgemeinen nicht vollständig)

FO Theorien

4. Sei K eine Klasse von τ-Stukturen. Dann ist

$$\mathsf{Th}(\mathcal{K}) = \bigcap_{\mathfrak{A} \in K} \mathsf{Th}(\mathfrak{A})$$

eine FO-Theorie (im allgemeinen nicht vollständig).

FO Theorien

Theorie Γ ist *entscheidbar* wenn folgendes Problem entscheidbar ist:

Gegeben: ein FO-Satz φ

Frage: ist $\varphi \in \Gamma$?

Wenn $\Gamma = \mathsf{Th}(\mathfrak{A})$, dann ist das also folgendes Problem:

Gegeben φ, entscheide ob $\mathfrak{A} \models \varphi$.

Das ist nicht Erfüllbarkeit / Gültigkeit, denn wir reden über eine feste Struktur anstatt über alle Strukturen.

Das ist nicht das bereits besprochene Modellprüfungsproblem, denn interessante $\mathfrak{A}$ sind unendlich!

Einige wichtige FO Theorien

Arithmetik der natürlichen Zahlen: $\mathsf{Th}(\mathbb{N}, +, *, 0, 1)$

Unentscheidbar und nicht rekursiv aufzählbar
(letzteres ist Gödels berühmter 1. Unvollständigkeitssatz)

Presburger Arithmetik: $\mathsf{Th}(\mathbb{N}, +, 0, 1)$

z.B. $\forall x\, ((x + x = x) \to x = 0)$

Entscheidbar, ungefähr 2ExpSpace-vollständig

Zu schwach, um wirklich interessante mathematische Probleme auszudrücken, aber wichtige Anwendungen in der Informatik!

Skolem Arithmetik: $\mathsf{Th}(\mathbb{N}, *, 0, 1)$

Entscheidbar, ungefähr 3ExpSpace-vollständig

Einige wichtige FO Theorien

Arithmetik der reellen Zahlen: $\mathsf{Th}(\mathbb{R}, +, *, 0, 1)$

Z.B. $\forall x \forall y \forall z \, ((x = y * y \land x = z * z) \rightarrow (y = z \lor y = z - (2 * z)))$

Entscheidbar, ungefähr ExpSpace-vollständig

Man vergrößert hier also den Zahlenbereich und bekommt dadurch ein einfacheres Entscheidungsproblem

Beachte:
Aus der Unentscheidbarkeit von $\mathsf{Th}(\mathbb{N}, +, *, 0, 1)$ folgt, dass man keine FO-Formel $\mathsf{Nat}(x)$ finden kann, die $\mathbb{N}$ in $\mathsf{Th}(\mathbb{R}, +, *, 0, 1)$ definiert.

FO Theorien

Interessanterweise hängt der Begriff der Vollständigkeit sehr eng mit der Definition von Theorien durch Strukturen zusammen.

Zwei τ-Strukturen $\mathfrak{A}$, $\mathfrak{A}'$ heissen *elementar äquivalent* wenn für alle Sätze $\varphi \in \mathsf{FO}(\tau)$ gilt: $\mathfrak{A} \models \varphi$ gdw. $\mathfrak{A}' \models \varphi$.

Lemma

Sei Γ eine FO-Theorie. Dann sind die folgenden Aussagen äquivalent:

1. Γ ist vollständig
2. $\Gamma = \mathsf{Th}(\mathfrak{A})$ für eine Struktur $\mathfrak{A}$
3. alle Modelle $\mathfrak{A}, \mathfrak{A}'$ von Γ sind elementar äquivalent

Axiomatisierung

Die Definition einer Theorie über eine Struktur verrät nicht viel über die enthaltenen Formeln:

Welche Sätze enthält Th$(\mathbb{Q}, <)$?

Die Definition über Abschluss(Π) ist viel aufschlussreicher, z.B.

$$
\begin{aligned}
\Pi = \{\ & \forall x\, \neg(x < x), && \text{strikte partielle} \\
& \forall x \forall y\, (x < y \rightarrow \neg y < x), && \text{Ordnung} \\
& \forall x \forall y \forall z\, (x < y \wedge y < z \rightarrow x < z), && \\
& \forall x \forall y\, (x < y \vee y < x), && \text{total} \\
& \forall x \exists y\, x < y,\ \forall x \exists y\, y < x, && \text{beidseitig unbeschr.} \\
& \forall x \forall y\, (x < y \rightarrow \exists z\, x < z < y)\ \} && \text{dicht}
\end{aligned}
$$

Es gilt Th$(\mathbb{Q}, <) =$ Abschluss(Π)!

Allgemein: Theorien *axiomatisieren*

Axiomatisierung

Definition Axiomatisierbar

Eine FO-Theorie Γ heisst *axiomatisierbar* wenn es eine entscheidbare Menge Π von Sätzen gibt, so dass $\Gamma = \mathsf{Abschluss}(\Pi)$, also

$$\Gamma = \{\varphi \text{ ist } \tau\text{-Satz} \mid \Pi \models \varphi\}.$$

Wenn Π endlich ist, heisst Γ *endlich axiomatisierbar*. Wir nennen Π eine (endliche) *Axiomatisierung* von Γ.

Beispiel: Die Theorie $\mathsf{Th}(\mathbb{Q}, <)$ ist endlich axiomatisierbar:

$$\begin{aligned}
\Pi = \{\ & \forall x\, \neg(x < x), \\
& \forall x \forall y\, (x < y \rightarrow \neg y < x), \\
& \forall x \forall y \forall z\, (x < y \wedge y < z \rightarrow x < z), \\
& \forall x \forall y\, (x < y \vee y < x), \\
& \forall x \exists y\, x < y,\ \forall x \exists y\, y < x, \\
& \forall x \forall y\, (x < y \rightarrow \exists z\, x < z < y)\ \}
\end{aligned}$$

Axiomatisierung

Weitere Beispiele:

Die Arithmetik $\mathtt{Th}(\mathbb{N}, +, *, 0, 1)$ ist nicht axiomatisierbar

Die Presburger Arithmetik $\mathtt{Th}(\mathbb{N}, +, 0, 1)$ ist axiomatisierbar, aber nicht endlich axiomatisierbar

Axiomatisierungen hängen eng zusammen mit Entscheidbarkeit:

Theorem

1. Eine FO-Theorie ist axiomatisierbar gdw. sie rekursiv aufzählbar ist.
2. Eine vollständige FO-Theorie ist axiomatisierbar gdw. sie entscheidbar ist.

Vergl. $\mathtt{Th}(\mathbb{N}, +, *, 0, 1)$: vollständig, unentscheidbar, nicht axiomatisierbar.