1 Motivation

Problemstellung

Gegeben: $\mathcal{A}x$, Menge von Axiomen.
φ, Formel.

Frage: Gilt φ in allen Modellen von $\mathcal{A}x$
(oder $\mathcal{A}x \models \varphi$)?

Beispiel: „Nilpotente Monoide sind kommutativ"

$$\begin{array}{ll} \mathcal{A}x: & \forall x.\ \ x \cdot 1 = x \\ & \forall x.\forall y.\forall z.\ \ (x \cdot y) \cdot z = x \cdot (y \cdot z) \\ & \forall x.\ \ x \cdot x = 1 \end{array}$$

$$\varphi: \quad \forall x.\forall y.\ \ x \cdot y = y \cdot x$$

Dann: Es gilt $\mathcal{A}x \models \varphi$.

Motivation (Forts.)

Beispiel Doktor-Quacksalber-Problem

a) Problemstellung verbal:
Einige Patienten mögen alle Doktoren.
Kein Patient mag einen Quacksalber.
Also ist kein Doktor ein Quacksalber.

b) Problemstellung formal:

$$\begin{array}{ll} \mathcal{A}x: & \exists x.(\mathrm{pat}(x) \wedge \forall y.(\mathrm{doc}(y) \Rightarrow \mathrm{mag}(x,y))) \\ & \forall x.(\mathrm{pat}(x) \Rightarrow \forall y.(\mathrm{qua}(y) \Rightarrow \neg\mathrm{mag}(x,y))) \end{array}$$

$$\varphi: \quad \forall x.(\mathrm{doc}(x) \Rightarrow \neg\mathrm{qua}(x))$$

Es gilt $\mathcal{A}x \models \varphi$.

Frage

Wie zeigt man dies?

Vorsicht

Das Problem ist unentscheidbar, aber rekursiv aufzählbar.

$\Rightarrow$ Der Suchraum ist i. Allg. unendlich.

$\Rightarrow$ Gute Suchheuristiken nötig.

Ansatz für Beweissysteme

1. Syntax + Semantik festlegen

Was ist eine Spezifikation?

Was heißt genau $\mathcal{A}x \models \varphi$?

2. Widerlegungsverfahren

Es gilt $\mathcal{A}x \models \varphi$ gdw $\mathcal{A}x \cup \{\neg\varphi\}$ ist unerfüllbar.

Zeige: Eine Menge Φ von Formeln ist unerfüllbar.

3. Transformation in Klauselform

Transformiere Φ in eine Menge S von Klauseln.

Sei $\square$ die leere Klausel.

4. Inferenzsystem auf Klauseln

Angabe von Inferenzsystem $\mathcal{I}$ mit

$S \overset{*}{\vdash}_{\mathcal{I}} \square$ gdw Φ ist unerfüllbar.

5. Transformation in einen Beweiser

Gute Datenstrukturen + Algorithmen.

Geschickte Architektur des Beweisers.

Suchraumeinschränkung.

Gute Suchheuristiken.

Voraussetzungen für die Vorlesung

Vorausgesetzt werden Grundkenntnisse aus der Logik

Aussagenlogik

- logische Operatoren
- Syntax + Semantik
- Folgerungsbegriff

Prädikatenlogik

a) Syntax
   Quantoren $\exists$ und $\forall$
   Aufbau der Terme und Formeln
   Spezifikationen

b) Semantik
   sig-Algebren $\mathcal{A}$
   Gültigkeit von Formeln in $\mathcal{A}$
   Gültigkeit in allen Modellen einer Spezifikation

Wiederholungen

Alle wesentliche Begriffe werden definiert und kurz motiviert.

Ziel

Effiziente Beweiser für die Prädikatenlogik 1. Stufe.

2 Prädikatenlogik 1. Stufe

2.1 Syntax und Semantik

Bemerkung

Beschränkung auf nur eine Sorte ANY (Die Verallgemeinerung auf mehrere Sorten ist einfach.)

Hilfssorte BOOL: Prädikatssymbole werden als BOOL-wertige Funktionssymbole behandelt.

Sei

F=Menge der Funktionssymbole.

P=Menge der Prädikatssymbole.

Stelligkeit $\alpha : F \cup P \to \mathbb{N}$ mit

$\alpha(f) = n$ heißt f : ANY$^n \to$ ANY für $f \in F$, $n \geq 0$
$\alpha(p) = n$ heißt p : ANY$^n \to$ BOOL für $p \in P$, $n \geq 0$

Definition 2.1.

Eine **Signatur** ist sig $= (F, P, \alpha)$ mit $F \cap P = \varnothing$,
$\alpha : F \cup P \to \mathbb{N}$ wie oben.

Bemerkung

Die $c \in F$ mit $\alpha(c) = 0$ heißen Konstanten.

Die $p \in P$ mit $\alpha(p) = 0$ heißen aussagenlogische Symbole.

Terme und Formeln (1)

Sei sig $= (F, P, \alpha)$ eine Signatur,
V= Menge von Variablen (der Sorte ANY).

Bezeichnungen

Die Menge Term(F, V) der **Terme** ist wie üblich definiert.

Ein **Atom** ist $p(t_1, \ldots, t_n)$ mit $p \in P$, $\alpha(p) = n$, t_i Terme.

Eine **Formel** φ ist

- ein Atom oder
- $\neg\varphi_1$, $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, $\varphi_1 \Rightarrow \varphi_2$, $\varphi_1 \Leftrightarrow \varphi_2$, $\exists x.\varphi_1$ oder $\forall x.\varphi_1$, falls φ_1, φ_2 Formeln sind.

Sei Form (F, P, V) die Menge aller Formeln.

Die Menge free(φ) der **freien Variablen** in φ ist

free(φ) =var(φ)	falls	φ Atom
free(φ_1)		$\varphi \equiv \neg\varphi_1$
free$(\varphi_1) \cup$ free(φ_2)		$\varphi \equiv \varphi_1 * \varphi_2$
free$(\varphi_1) \backslash \{x\}$		$\varphi \equiv \exists x.\varphi_1$ oder $\varphi \equiv \forall x.\varphi_1$

Hier ist $* \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$.

Eine Formel φ heißt **abgeschlossen**, falls free$(\varphi) = \varnothing$ gilt.

Terme und Formeln (2)

Bezeichnungen

Die Menge Pos(φ) der **Positionen** in φ ist

- wie üblich, falls φ Atom.
- $\{1.\pi \mid \pi \in \text{Pos}(\varphi_1)\} \cup \{\lambda\}$, falls $\varphi \equiv \neg\varphi_1, \forall x.\varphi_1, \exists x.\varphi_1$
- $\{1.\pi \mid \pi \in \text{Pos}(\varphi_1)\} \cup \{2.\pi \mid \pi \in \text{Pos}(\varphi_2)\} \cup \{\lambda\}$,
  falls $\varphi \equiv \varphi_1 * \varphi_2$

Ein **Literal** l ist A oder $\neg A$, wobei A ein Atom ist.

Eine **Klausel** $C \equiv l_1 \vee \cdots \vee l_n$ ist eine Disjunktion von Literalen.

Schreibweisen

$\forall x_1, \ldots, x_n.\varphi$ statt $\forall x_1. \cdots \forall x_n.\varphi$

$C = \{l_1, \ldots, l_n\}$ statt $C \equiv l_1 \vee \cdots \vee l_n$

$\bar{x}_n$ statt $x_1, \ldots, x_n$ und $\bar{t}_n$ statt $t_1, \ldots, t_n$

$\varphi_1 \vee \varphi_2 \vee \varphi_3$ ohne Klammern, da $\vee$ assoziativ.

Also

Klauseln sind Multimengen von Literalen.

Semantik (1)

Sei sig $= (F, P, \alpha)$ eine Signatur,
$V =$ Menge von Variablen.

Definition 2.2.

Eine **sig-Algebra** $\mathcal{A} = (A;\ F^{\mathcal{A}}, P^{\mathcal{A}})$ besteht aus
$A \neq \varnothing$, eine Menge (das Universum),
$F^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in F\}$ mit $f^{\mathcal{A}} : A^n \to A$, falls $\alpha(f) = n$,
$P^{\mathcal{A}} = \{p^{\mathcal{A}} \mid p \in P\}$ mit $p^{\mathcal{A}} \subseteq A^n$, falls $\alpha(p) = n$.

Intention

Ein Funktionssymbol $f \in F$ wird durch eine Funktion $f^{\mathcal{A}}$ interpretiert.

Ein Prädikatssymbol $p \in P$ wird durch ein Prädikat $p^{\mathcal{A}}$ interpretiert. Wir identifizieren ein n-stelliges Prädikat $p^{\mathcal{A}}$ auf A mit einer Teilmenge von A^n und einer n-stelligen Relation auf A.

Die Variablen $x \in V$ werden durch eine **Evaluierungsfunktion** eval : $V \to A$ interpretiert.

Semantik (2)

Sei sig $= (F, P, \alpha)$ eine Signatur,
$(\mathcal{A}, \text{eval})$ eine sig-Algebra mit Evaluierungsfunktion.

Definition 2.3.

Der **Wahrheitswert** $\text{eval}(\varphi) = \text{eval}_{\mathcal{A}}(\varphi) \in \{\text{true}, \text{false}\}$ einer Formel φ in $\mathcal{A}$ ist rekursiv erklärt durch

- $\text{eval}(c) = c^{\mathcal{A}}$
- $\text{eval}(f(t_1, \ldots, t_n)) = f^{\mathcal{A}}(\text{eval}(t_1), \ldots, \text{eval}(t_n))$
- $\text{eval}(p(t_1, \ldots, t_n)) = \text{true gdw } (\text{eval}(t_1), \ldots, \text{eval}(t_n)) \in p^{\mathcal{A}}$
- $\text{eval}(\neg\varphi) = \neg\text{eval}(\varphi)$
- $\text{eval}(\varphi_1 * \varphi_2) = \text{eval}(\varphi_1) * \text{eval}(\varphi_2)$ für $* \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$
- $\text{eval}(\forall x.\varphi) = \text{true gdw}$ für alle $a \in A$ gilt $\text{eval}^{x,a}(\varphi) = \text{true}$
- $\text{eval}(\exists x.\varphi) = \text{true gdw}$ für ein $a \in A$ gilt $\text{eval}^{x,a}(\varphi) = \text{true}$

Dabei ist

$$\text{eval}^{x,a}(y) = \begin{cases} a & y = x \\ \text{eval}(y) & y \neq x \end{cases}$$

Semantik (3)

Lemma 2.1

Seien φ abgeschlossen, $\mathcal{A}$ eine sig-Algebra und $\text{eval}, \text{eval}'$ Evaluierungsfunktionen.

Dann gilt $\text{eval}(\varphi) = \text{eval}'(\varphi)$.

Bezeichnung

$\mathcal{A} \models \varphi$, falls für alle Evaluierungsfunktionen $\text{eval}_{\mathcal{A}}$ gilt $\text{eval}_{\mathcal{A}}(\varphi) = \text{true}$.

Definition 2.4.

Eine **Spezifikation** spec $= (\text{sig}, \mathcal{A}x)$ besteht aus einer Signatur sig und einer Menge $\mathcal{A}x$ von abgeschlossenen Formeln (Axiomen).

Eine sig-Algebra $\mathcal{A}$ ist ein **Modell** von spec, falls $\mathcal{A} \models \varphi$ für alle $\varphi \in \mathcal{A}x$ gilt.

Bezeichnung

Mod(spec) ist die Klasse aller Modelle von spec.

Definition 2.5.

a) φ heißt **erfüllbar**, falls es eine sig-Algebra $\mathcal{A}$ gibt mit $\mathcal{A} \models \varphi$.
b) $\mathcal{A}x \models \varphi \text{ gdw } \mathcal{A} \models \varphi$ für alle $\mathcal{A} \in \text{Mod}(\text{sig}, \mathcal{A}x)$.

Beispiele

Beispiel 1

Sei $F = \emptyset$, $P = \{p\}$, $\varphi \equiv \forall x.\exists y.p(x, y)$, $A = \mathbb{N}$.

a) $\mathcal{A} = (\mathbb{N};\ P^{\mathcal{A}})$ mit $p^{\mathcal{A}}(x, y)$ gdw $x < y$.
   Es gilt $\mathcal{A} \models \varphi$, denn zu jedem $x \in \mathbb{N}$ gibt es ein $y \in \mathbb{N}$ mit $x < y$.

b) $\mathcal{A} = (\mathbb{N};\ P^{\mathcal{A}})$ mit $p^{\mathcal{A}}(x, y)$ gdw $x > y$.
   Es gilt nicht $\mathcal{A} \models \varphi$, denn für $x = 0$ gibt es kein $y \in \mathbb{N}$ mit $x > y$.

Beispiel 2

Sei $F = \emptyset, P = \{p\}$, $\mathcal{A}x = \{\varphi_1, \varphi_2, \varphi_3\}$ mit

- $\varphi_1 \equiv \forall x.p(x, x)$
- $\varphi_2 \equiv \forall x, y.(p(x, y) \Rightarrow p(y, x))$
- $\varphi_3 \equiv \forall x, y, z.(p(x, y) \wedge p(y, z) \Rightarrow p(x, z))$

Sei spec = (sig, $\mathcal{A}x$).

Eine sig-Algebra $\mathcal{A}$ ist genau dann ein Modell von spec, wenn $p^{\mathcal{A}}$ eine Äquivalenzrelation ist.

Prädikatenlogik mit Gleichheit: Motivation

Bisher

Beschrieben ist die Prädikatenlogik 1. Stufe.

Bezeichnung: PL1 oder FOL.

Beachte: Jedes $p \in P$ kann beliebig interpretiert werden.

Jetzt

Prädikatenlogik 1. Stufe mit Gleichheit.

Bezeichnung: PL1= oder FOLE.

P enthält das Prädikatssymbol „=“ mit $\alpha(=) = 2$, es wird stets als das Gleichheitsprädikat interpretiert.

Beachte

$=$ muss in $\mathcal{A}$ stets als eine Kongruenzrelation interpretiert werden. Es muss z. B. gelten

$$\mathcal{A} \models t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \Rightarrow f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)$$

$$\mathcal{A} \models t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge p(t_1, \ldots, t_n) \Rightarrow p(t'_1, \ldots, t'_n)$$

Prädikatenlogik mit Gleichheit: Formal

1. Ansatz: PL1=

Es bleibt alles wie in PL1, aber

1. $= \in P$ mit $\alpha(=) = 2$
2. bei festem $(\mathcal{A}, \text{eval})$ gilt
   $\text{eval}(t_1 = t_2) = \text{true}$ gdw $\text{eval}(t_1) = \text{eval}(t_2)$ in $\mathcal{A}$.

Beachte

Hier wird $=$ stets fest interpretiert (als Gleichheit).

Frage

Kann man PL1= in PL1 einbetten?

Einbettung von PL1= in PL1

2.Ansatz für PL1

Bezeichnung

$\mathcal{A}x_=$ bestehe aus (für alle $f \in F$, $p \in P$)

$$\forall x.\ x = x$$

$$\forall x, y.\ (x = y \Rightarrow y = x)$$

$$\forall x, y, z.\ (x = y \wedge y = z \Rightarrow x = z)$$

$$\forall x_1, y_1, \ldots, x_n, y_n.(x_1 = y_1 \wedge \cdots \wedge x_n = y_n \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n))$$

$$\forall x_1, y_1, \ldots, x_n, y_n.(x_1 = y_1 \wedge \cdots \wedge x_n = y_n \wedge p(x_1, \ldots, x_n) \Rightarrow p(y_1, \ldots, y_n))$$

Satz 2.1.

Für jede Spezifikation $\text{spec} = (\text{sig}, \mathcal{A}x)$ gilt

$\mathcal{A}x$ ist bezüglich PL1= erfüllbar gdw
$\mathcal{A}x \cup \mathcal{A}x_=$ ist bezüglich PL1 erfüllbar.

Beispiel zu PL1=

Sei

spec = (sig, $\mathcal{A}x$) mit

$$F = \{1, {}^{-1}, *\},\ \ P = \{=\},\ \ \mathcal{A}x = \{\varphi_1, \varphi_2, \varphi_3\}$$
$$\varphi_1 \equiv \forall x.\ \ x * 1 = x$$
$$\varphi_2 \equiv \forall x.\ \ x * x^{-1} = 1$$
$$\varphi_3 \equiv \forall x, y, z.\ \ (x * y) * z = x * (y * z)$$

Dann

Eine sig-Algebra $\mathcal{A}$ ist genau dann ein Modell von spec, wenn $\mathcal{A}$ eine Gruppe ist.

Beispiel $\mathcal{A} = (A, F^{\mathcal{A}}, P^{\mathcal{A}})$

a) $A = \mathbb{N},\quad 1^{\mathcal{A}} = 0,\quad {}^{-1\mathcal{A}}$=id, $\quad *^{\mathcal{A}} = +$
   $\mathcal{A}$ ist kein Modell von spec.
b) $A = \mathbb{Z},\quad 1^{\mathcal{A}} = 0,\quad {}^{-1\mathcal{A}} = -,\quad *^{\mathcal{A}} = +$
   $\mathcal{A}$ ist ein Modell von spec.
c) $A = \{0,1\},\quad 1^{\mathcal{A}} = 0,\quad {}^{-1\mathcal{A}}$=id, $\quad *^{\mathcal{A}} = +$ modulo 2
   $\mathcal{A}$ ist ein Modell von spec.

2.2 Transformation in Klausel-Normalform

Bezeichnung

Eine Formel φ ist in **Klausel-Normalform (CNF)**, falls sie die Form hat

$$\varphi \equiv \forall^*(C_1 \wedge \cdots \wedge C_n)$$

wobei die C_i Klauseln sind.

Ziel (*)

Transformation von φ in φ' mit

(1) φ' ist in CNF.

(2) φ ist erfüllbar gdw φ' ist erfüllbar.

Dann heißt φ' eine **CNF für** φ.

Beachte

Ist φ' in CNF, so enthält φ' nicht den Quantor $\exists$ und die C_i sind ganz Quantoren-frei.