Allgemeingültigkeit, Erfüllbarkeit

Eine Struktur $S = [U, I]$

- beschreibt eine Menge U von Objekten: Individuenbereich, Universum
- legt mit der Interpretation I die Bedeutung von Funktionszeichen und Prädikatenzeichen über U fest.

Eine Belegung β ordnet jeder Variablen ein Objekt aus U zu.

Es gilt $Wert_{[U,I]}(H, \beta) = W$, falls der Ausdruck H in der Struktur $S = [U, I]$ bei der Belegung β wahr ist.

Allgemeingültigkeit, Erfüllbarkeit

Belegung β erfüllt den Ausdruck H in der Struktur $S = [U, I]$, falls $Wert_{[U,I]} (H, \beta) = W$.

(Rein logische) Erfüllbarkeit eines Ausdrucks H:
H heißt erfüllbar, falls β, U, I existieren mit $Wert_{[U,I]} (H, \beta) = W$.

ef : Menge aller erfüllbaren Ausdrücke

(Rein logische) Allgemeingültigkeit eines Ausdrucks H:
H heißt allgemeingültig, falls $Wert_{[U,I]} (H, \beta) = W$ für alle β, U, I .

ag : Menge aller allgemeingültigen Ausdrücke

Freie Variable werden bei Allgemeingültigkeit wie generalisierte Variable behandelt.

Allgemeingültigkeit, Erfüllbarkeit

H ist allgemeingültig gdw. $\neg H$ nicht erfüllbar ist.
H ist erfüllbar gdw. $\neg H$ nicht allgemeingültig ist.
ag axiomatisierbar:
Es gibt aufzählbares Axiomensystem **axp** mit **ag** = Abl(**axp**)
Satz von Church:
Erfüllbarkeit/Allgemeingültigkeit sind unentscheidbar.

Menge der allgemeingültigen Ausdrücke: axiomatisierbar.
Menge der nicht erfüllbaren Ausdrücke: axiomatisierbar.

Menge der nicht allgemeingültigen Ausdrücke: nicht axiomat.
Menge der erfüllbaren Ausdrücke: nicht axiomatisierbar.

Axiomatisierbar = aufzählbar = partiell entscheidbar
M entscheidbar gdw. M und U-M aufzählbar

Folgern im PK1

Eine Struktur $S = [U, I]$ und eine Belegung β sind ein *Modell* für eine Menge **X** von Ausdrücken, wenn für alle $H \in$ **X** gilt:
β erfüllt H in der Struktur $S = [U, I]$, d.h. $Wert_{[U,I]}(H, \beta) = W$.

Es sei **X** eine Menge von Ausdrücken, H ein Ausdruck.
H *folgt aus* **X** , falls gilt:
Jedes Modell von **X** ist ein Modell von H .

$\mathbf{X} \models H$ oder: $H \in \mathbf{X}|{=}$ oder: $H \in Fl(\mathbf{X})$

Folgern im PK1

Fl ist syntaktisch beschreibbar mittels Abl
Fl = Abl „modulo **ag**“

Fl und Abl sind monoton:
$\mathbf{X} \subseteq \mathbf{Y} \Rightarrow Fl(\mathbf{X}) \subseteq Fl(\mathbf{Y})$
$\mathbf{X} \subseteq \mathbf{Y} \Rightarrow Abl(\mathbf{X}) \subseteq Abl(\mathbf{Y})$

$\mathbf{ag} = Fl(\emptyset) = Abl(\mathbf{axp})$

$(H \rightarrow G) \in Fl(\mathbf{X}) \Leftrightarrow G \in Fl(\mathbf{X} \cup \{H\})$

$H \in Fl(\mathbf{X}) \Leftrightarrow (\wedge \mathbf{X} \rightarrow H) \in \mathbf{ag}$

Formale Theorien im PK1

Als *Theorie* wird eine bezüglich Folgern (Ableiten) abgeschlossene Menge **Th** von Ausdrücken bezeichnet:
$\mathbf{Th} = Fl(\mathbf{Th}) = Abl(\mathbf{Th} \cup \mathbf{ag})$

Theorie mit *semantisch bestimmter Satzmenge*:
Gegeben ist eine Struktur $S = [U, I]$ mit
$\mathbf{Th} = \{ F \mid F$ allgemeingültig in $S\}$
$= \{ F \mid Wert_{[U,I]} (F, \beta) = W$ für alle β über $U \}$
Theorie mit *syntaktisch bestimmter Satzmenge*:
Gegeben ist ein Ausdruckmenge **Ax** („Axiome“) mit
$\mathbf{Th} = Abl(\mathbf{Ax} \cup \mathbf{axp})$ ($= Fl(\mathbf{Ax})$)

Formalisierung mittels PK1

Ziel: Axiome zur Beschreibung von Sachverhalten finden
Analoges Ziel: **Computerverarbeitung ermöglichen**

1. semantisch definierte Theorie **Th** bestimmen
   (Universum, Relationen/Funktionen, Interpretation)
2. axiomatische Beschreibung von **Th** :
   Axiome **Ax** mit **Th** = Abl(**Ax** $\cup$ **axp**)

   (z.B. PROLOG-Programm)

Anwendung PK1

Formalisierung: Axiome **X**
Problem durch Ausdruck H beschreiben
Entscheiden, ob

$H \in Fl\ (\mathbf{X})$

d.h. ob $H \in Abl(\mathbf{X} \cup \mathbf{axp})$

Programme dafür: Theorembeweiser

Beweise

– Positiver Kalkül: Allgemeingültigkeit entscheiden

$\boxed{H \in \mathbf{Th}\ ?}$

– Negativer Kalkül: Unerfüllbarkeit untersuchen

$\boxed{\{\neg H\} \cup \mathbf{Th} \text{ widersprüchlich ?}}$

– Deduktiver Kalkül:
  Erweitern der Axiome um H zu finden (foward chaining)

– Testkalkül:
  Reduktion von H auf Axiome (backward chaining)

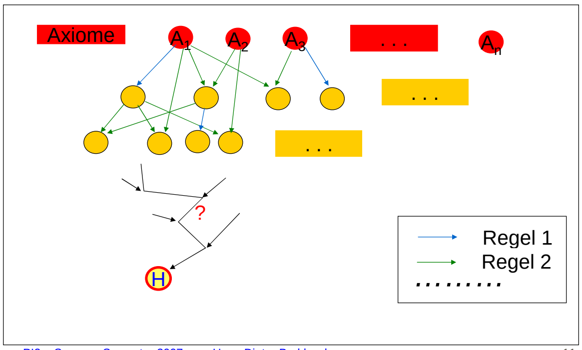
Beispiel: Resolution (PROLOG) ist *Negativer Testkalkül*

Beweise

$H \in Fl\ (\mathbf{X})$
gilt gdw. $H \in Abl(\mathbf{X} \cup \mathbf{axp})$ ⇒ Positiver Kalkül

$H \in Fl\ (\mathbf{X})$
gilt gdw. $(\wedge\mathbf{X} \rightarrow H) \in \mathbf{ag}$
gilt gdw. $\neg(\wedge\mathbf{X} \rightarrow H) \notin \mathbf{ef}$
gilt gdw. Skolemform von $(\wedge\mathbf{X} \wedge \neg H)$ nicht erfüllbar
gilt gdw. Klauselform von $(\wedge\mathbf{X} \wedge \neg H)$ nicht erfüllbar
⇒ Negativer Kalkül

Positiver deduktiver Kalkül



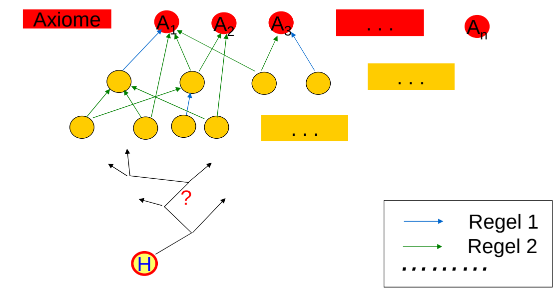
Suchraum bei deduktivem Kalkül

Klassischer AK : 15 Axiome für **ag**, 2 Regeln
(ist entscheidbar, allerdings NP)

Klassischer PK1:
– abzählbar viele Axiome für **ag** + weitere für **Th** ,
– 7 Regeln

Vollständigkeit als Nachteil:
– Alle allgemeingültigen Ausdrücke im Suchraum **Th** :
  $\mathbf{ag} = Fl(\varnothing) \subseteq Fl(\mathbf{Th}) = \mathbf{Th}$
– Mit H viele weitere Ausdrücke im Suchraum **Th** :
  z.B. $H \vee G$ für beliebiges G

Positiver Test-Kalkül



Normalformen

ein Ausdruck H heißt *bereinigt*, falls gilt:

- keine Variable x kommt in H sowohl frei als auch gebunden vor,
- hinter den Quantoren in H vorkommende Variable sind verschieden.

ein Ausdruck H heißt *pränex*, falls gilt:

- H hat die Form $Q_1 x_1 \dots Q_n x_n H'$,
  wobei $Q_1, \dots, Q_n$ Quantoren sind und H´ keine Quantoren enthält.

ein Ausdruck H heißt *pränexe Normalform*, falls gilt:

- H hat pränexe Form $Q_1 x_1 \dots Q_n x_n H'$,
- H´ ist eine konjunktive Normalform (KNF):

$$H' = \bigwedge \{ \{ \bigvee L_{ij} \mid i=1\dots n \} \mid j=1\dots m_n \}$$

(die L_{ij} sind Literale)

Literal: atomare Formel oder negierte atomare Formel

Normalformen

Sätze

1. Zu jedem Ausdruck H_1 existiert ein Ausdruck H_2 in bereinigter Form.
2. Zu jedem Ausdruck H_2 in bereinigter Form existiert ein semantisch äquivalenter Ausdruck H_3 in bereinigter pränexer Form
3. Zu jedem Ausdruck H_3 in bereinigter pränexer Form existiert semantisch äquivalenter Ausdruck H_4 in bereinigter pränexer Normalform.

Zu jedem Ausdruck H existiert ein semantisch äquivalenter Ausdruck G in bereinigter pränexer Normalform.

Skolem-Normalform

Eine Skolem-Normalform ist eine pränexe Normalform, deren Variable sämtlich generalisiert sind.

Umformung einer bereinigten pränexen Normalform G in eine Skolem-Normalform F:

1. ∃-Quantoren einführen für alle freien Variablen in G.
   (erfüllbarkeits-äquivalente Umformung !!)
2. Elimimation aller ∃-Quantoren durch Skolem-Funktionen.

Die entstehende Formel F ist erfüllbarkeits-äquivalent zu G.

Klauselform

Eine Klausel ist eine Menge von Literalen $K = \{L_1, \dots, L_n\}$.
Vorteil der Klauselnotation: Mengenschreibweise beseitigt rein formale Unterschiede äquivalenter Ausdrücke bzgl. Kommutativität/Idempotenz.

Umformung einer Skolem-Normalform F in Klauselform:

1. Verteilung der Allquantoren auf die Alternativen.
   (Semantisch äquivalente Umformung)
2. Gebundenene Umbenennungen derart, dass sich insgesamt alle Quantoren auf unterschiedliche Variablen beziehen.
   (Semantisch äquivalente Umformung)
3. Darstellung der Alternativen als Klauseln (Menge von Literalen).
   Darstellung von F als Menge von von Klauseln **Kl** :

$$\mathbf{Kl} = \{ \{ L_{ij} \mid j = 1, \dots m_i \} \mid i = 1, \dots, n \}$$

Klauselform

A1: $\forall x \neg R(x,x)$

A2: $\forall x \forall y \forall z (R(x,y) \wedge R(y,z) \rightarrow R(x,z))$

$\forall x \forall y \forall z (\neg(R(x,y) \wedge R(y,z)) \vee R(x,z))$

$\forall x \forall y \forall z (\neg R(x,y) \vee \neg R(y,z) \vee R(x,z))$

Klauseln für A1:

KA1: $\neg R(x,x)$

Klauseln für A2:

KA2: $\neg R(u,y) \vee \neg R(y,z) \vee R(x,u))$

Klauselform

Negation bei Resolutionsbeweis

H: $\forall x\ \forall y\ (R(x,y) \rightarrow \neg R(y,x))$

$\neg$H: $\neg\forall x\ \forall y\ (R(x,y) \rightarrow \neg R(y,x))$

$\exists x\ \exists y\ \neg(\neg R(x,y) \vee \neg R(y,x))$

$\exists x\ \exists y\ (R(x,y) \wedge R(y,x))$

$R(c,f(c)) \wedge R(f(c),c)$

Klauseln für $\neg$H:

K1: $R(c,f(c))$

K2: $R(f(c),c)$

Substitution

Substitution σ :

Abbildung der Individuenvariablen in die Menge der Terme.

Variablenumbenennung: Ersetzung von Variablen durch Variable.

$\sigma(L)$: Ergebnis der Substitution σ für ein Literal (rekursiv definiert).

Hintereinanderausführung von Substitutionen:

$\sigma_1 * \sigma_2 (x) = \sigma_2(\sigma_1(x))$

$\sigma(L)$ ist jeweils „Spezialfall" von L .

L ist die allgemeinste Form bzgl. aller $\sigma(L)$ für beliebige σ.

Substitution, Unifikation

Eine Substitution σ heißt *Unifikator*
für eine Menge $\{L_1, ..., L_n\}$ von Literalen,
falls gilt: $\sigma(L_1) = \sigma(L_2) = ...= \sigma(L_n)$
(syntaktische Gleichheit).

Ein Unifikator σ heißt *allgemeinster Unifikator*
(m.g.u. = most general unifier), falls für jeden Unifikator σ_0 eine Substitution σ_{00} existiert mit

$\sigma_0 = \sigma * \sigma_{00}$ (σ_0 ist spezieller als σ).

Der allgemeinste Unifikator

ist eindeutig bis auf Umbenennungen.

Substitution, Unifikation

Der allgemeinste Unifikator σ einer Menge $\{ L_1,...,L_n \}$ von Literalen beschreibt die Menge der gemeinsamen Spezialisierungen (Beispiele).

Werden die Variablen der Literale zuvor separiert, so ergibt sich ein allgemeinerer Unifikator.

Satz

Es ist entscheidbar, ob ein Unifikator existiert.

Zu jeder unifizierbaren Menge von Literalen existiert ein allgemeinster Unifikator.

Ein allgemeinster Unifikator kann ggf. konstruiert werden.

Resolutionsregel

$$\begin{tikzcd}[column sep=small] K1 \arrow[dr] & & K2 \arrow[dl] \\ & K & \end{tikzcd}$$

K_1 und K_2 Klauseln ohne gemeinsame Variablen
(sonst: Separation durch Variablenumbenennungen)

Literale $L'_1,...,L'_{n'} \in K_1$ und $M'_1,...,M'_{m'} \in K_2$, $n',m' > 0$,
die mittels eines Unifikators σ unifizierbar sind.

Durch Resolution von K_1 und K_2 entsteht die *Resolvente*

$K = \sigma((K_1 - \{L'_1,...,L'_{n'}\}) \cup (K_2 - \{M'_1,...,M'_{m'}\}))$

$$\frac{\{L_1,..,L_n,L'_1,..,L'_{n'}\},\ \{M_1,..,M_m,M'_1,..,M'_{m'}\},\ \ \sigma(L'_1)=...=\sigma(L'_n) = \sigma(\neg M'_1)=...=\sigma(\neg M'_m)}{\sigma(\{ L_1,..,L_n , M_1,..,M_m \}\ )}$$

Resolutionsregel

Zerlegung in zwei Regeln.

Einfache Resolutionsregel:

$$\frac{\{L_1,..,L_n,L'\},\ \{M_1,.., M_m,M'\},\ \sigma(L')=\sigma(\neg M')}{\sigma(\{ L_1,..,L_n , M_1,..,M_m \}\ )}$$

Faktorisierungsregel:

$$\frac{\{L_1,..,L_n,L_1',L_2'\},\ \sigma(L_1')=\sigma(L_2')}{\sigma(\{ L_1,..,L_n ,L_1' \})}$$

Spezialfälle

Schnittregel

$$\frac{A \vee B,\ \neg B' \vee C\ ,\ \sigma(B)=\sigma(B')}{\sigma\,(A \vee C)}$$

Modus ponens

$$\frac{B,\ B \rightarrow C}{C}$$

Indirekter Beweis

$$\frac{B,\ \neg\, B}{\ }$$