Prädikatenlogik

- Syntax
- Semantik
- Formeln, Modelle, Tautologien und Anwendungen
- Entscheidbarkeitsfragen
- Kalküle und Tableaux
- Normalformen und Resolution

Prädikatenlogik

- **Syntax**

Von der Aussagenlogik zur Prädikatenlogik (1)

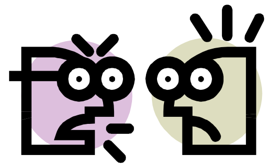
In der Aussagenlogik ...

betrachten wir **globale Aussagen über ganze Systeme**.

Aussagen sind **je nach betrachtetem System** (konkretes System bzw. abstraktes System = Belegung der Aussagevariablen) **wahr oder falsch**. **Keine** logische Behandlung **interner Details** der Systeme.

Was ist aber mit (logischen?) Argumentationen wie ...

„**Wer jemanden** anmeckert, fliegt raus!“ –
„Aber **Anna** hat **mich** doch gerade angemeckert
und ist nicht rausgeflogen!“



„**Gibt es** Adverbien **die** mit S anfangen?“ –
„Sicher!“

..., bei denen es **im betrachteten System** um Eigenschaften oder Beziehungen einzelner oder aller **Individuen** einer **Population** geht?

Von der Aussagenlogik zur Prädikatenlogik (2)

Prädikatenlogik:

die Welt, die ganzen Zahlen, ...

Wir **schauen in die Systeme hinein** und befassen uns mit

- **Objekten** innerhalb eines Systems sowie ihren möglichen
- **Eigenschaften** und
- **Beziehungen** zueinander.

Aussagen sprechen vom **Zutreffen von Eigenschaften** oder **Beziehungen** auf **einzelne** oder **alle Objekte** innerhalb des Systems.

Ausnahmen?

Aussagen sind in der Regel **je nach**

- betrachtetem **System und** den
- bezeichneten **Objekten** und **Eigenschaften** oder **Beziehungen**

wahr oder falsch.

Der Weg zur PL-Syntax und -Semantik (1)

Syntax: AL + zusätzliche **Ausdrucksmittel zur Benennung** von

- bestimmten oder beliebigen **Objekten**:

  → Eigennamen (*Konstanten*, *Variablen*), Terme (auch mit *Funktionen*)

- **Eigenschaften** von Objekten oder
  Beziehungen zwischen mehreren Objekten

  → *Prädikate*, *Relationen*

Syntaxbeispiel:

sprachlich: Zu jedem Objekt existiert ein echt größeres.

symbolisch: $\boxed{\forall x \exists y G(y, x)}$

Der Weg zur PL-Syntax und -Semantik (2)

Semantik: Interpretation der Objekt- und der Eigenschaften-Namen in einem konkreten System.

Semantikbeispiele:

Die obige Aussage ist

- **wahr**, wenn

  > Objekte := Intervalle [r,s] in den reellen Zahlen, mit [a,b] größer als [c,d] :⇔ $b - a > d - c$;

- **falsch**, wenn

  > Objekte := Münzen in meiner Geldbörse, Größe ≅ Durchmesser

„Alphabet“ der Prädikatenlogik erster Stufe („PL1“)

Wir definieren das **Alphabet**, die **Terme** (für Objekte) und dann **PL1-Formeln**.

Das **Alphabet** der Prädikatenlogik erster Stufe PL1 umfasst

- eine abzählb. Menge *VS* von **(Objekt-) Variablensymbolen**,
  hier $x_1, x_2, ...$
  – informell *u*, *v*, *w*, *x*, *y*, *z*, *Mitarbeiter* usw.
- eine abzählb. Menge *KS* von **(Objekt-) Konstantensymbolen**,
  hier $a_1, a_2, ...$.
  – informell *a*, *b*, *c*, ... , *Besitzer* usw.
- eine abzählbare Menge *FS* von **Funktionssymbolen**, hier f_1, f_2, ...
  – informell *f*, *g*, *h*, ... , *Vorgesetzte(r)*, usw. –
  – mitsamt Abbildung (F-)Stelligkeit : $FS \rightarrow \mathbb{N}$
- eine abzählbare Menge *PS* von **Prädikat-(Relations-)symbolen**, hier
  P_1, P_2, ...
  – informell *P*, *Q*, *R*, ... , *arbeitet_für*, usw. –
  – mitsamt Abbildung (P-)Stelligkeit : $PS \rightarrow \mathbb{N}$

Alphabet von PL1 (Rest)

- **Junktoren**

  $\neg \;\wedge\; \vee\; \rightarrow\; \leftrightarrow$

- **Quantoren**

  $\forall$ *für alle* ... *gilt* ... **Allquantor, Generalisierung**

  $\exists$ *es existiert (mindestens) ein* ... *so dass* ... **Existenzquantor, Partikularisierung**

- **syntaktische Hilfssymbole** Klammern ( ) und Komma , .

auch mal als **{ }** oder **[ ]**

Sprache der Prädikatenlogik PL1: Terme

Mit **Termen** benennen wir **Objekte**.

Die PL1-Terme sind **induktiv** so definiert:

- Alle (Objekt-) **Konstanten** und **Variablen** sind **Terme**;
- Ist f ein k-stelliges **Funktion**ssymbol ($k \geq 1$) und sind $t_1, \ldots, t_k$ **Terme**, so ist $f(t_1, \ldots, t_k)$ ein **Term**.

Alternativer PL1-Dialekt:

Objektkonstanten ≅ *„nullstellige" Funktionen*

Sprache der Prädikatenlogik PL1: Formeln

Die Aussagen der PL1 heißen **(PL1)-Formeln**.

PL1-Formeln sind **induktiv** so definiert:

- Ist P ein k-stelliges Relationssymbol ($k \geq 1$) und sind $t_1, \ldots, t_k$ Terme, so ist $P(t_1, \ldots, t_k)$ eine (**atomare**) Formel;

  alternativer PL1-Dialekt: *nullst. Rel.n ≅ „Aussagenvariablen“*

- Sind F und G Formeln, dann auch $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$, und $(F \leftrightarrow G)$.

- Ist F eine Formel und x eine Variable, so sind $\forall xF$ und $\exists xF$ Formeln.

komplexe Formeln: nicht atomar,

Teilformeln, **Teilterme** von Formeln
Induktion/Rekursion über Formelaufbau
Formel- und/oder **Termbäume**

analog AL

evtl. Klammern
wg. Lesbarkeit, z.B.

$\forall xP(x) \rightarrow \exists yQ(x,y)$ →

$\forall x[P(x) \rightarrow \exists yQ(x,y)]$??

$\forall x[P(x)] \rightarrow \exists y[Q(x,y)]$!!

PL1 und natürliche Sprache: Übung

Darmstadt ist eine Großstadt südlich von Frankfurt.

Keine deutsche Großstadt liegt nördlich von Flensburg.

Alle Menschen werden Brüder.

Vorsicht – Sprachfeinheiten!

PL1 und natürliche Sprache: Lösung

Darmstadt ist eine Großstadt südlich von Frankfurt(M).

da := Darmstadt
f := Frankfurt(M)
$Gr(x) :\Leftrightarrow x$ ist Großstadt
$S(x, y) :\Leftrightarrow x$ liegt südlich von y

$Gr(da) \wedge S(da, f)$

Keine deutsche Großstadt liegt nördlich von Flensburg.

fl := Flensburg
$Gr(x) :\Leftrightarrow x$ ist Großstadt
$N(x, y) :\Leftrightarrow x$ liegt nördlich von y

$\neg \exists x\ (Gr(x) \wedge N(x, fl))$

Alle Menschen werden Brüder.

Vorsicht – Sprachfeinheiten!

$M(x) :\Leftrightarrow x$ ist Mensch.
$Br(x, y) :\Leftrightarrow x$ <u>wird</u> Bruder von y

$x \neq y$?!

Zeit? Veränderung?

$\forall x \forall y\ \ (x \neq y \wedge (M(x) \wedge M(y))) \rightarrow Br(x, y))$

PL1 und natürliche Sprache: Grundmengen, Konstanten

- Wovon wird gesprochen?
  D.h. wer sind die Objekte bzw. die **Grundmengen**?
- Die hatten wir stillschweigend vorausgesetzt.
- Teils verschiedene sinnvolle Möglichkeiten:

Darmstadt ist eine Großstadt südlich von Frankfurt.

Keine deutsche Großstadt liegt nördlich von Flensburg.

→ **Grundmenge** (deutsche) (Groß-)Städte, Örtlichkeiten auf der Erde, …

→ **Konstanten**: Darmstadt, Frankfurt, Flensburg

Alle Menschen werden Brüder.

→ **Grundmenge** Menschen, Lebewesen, Gegenstände und Lebewesen, …

→ **Konstanten**: –

PL1 und natürliche Sprache: gemischte Populationen, Funktionen

Deine Meinung interessiert mich nicht.

Grundmenge
Konstanten
Funktionen
Relationen

Brunos Vater ist ein Vorgesetzter von Stefanies Mutter.

PL1 und natürliche Sprache: gemischte Populationen, Funktionen

Deine Meinung interessiert mich nicht.

Grundmenge: Personen + Meinungen
Konstanten: *ich*, *du* Personen
Funktionen: $m(x)$ Meinung von x (*mit z.B.* $m(w)=w/\varnothing$ für Meinungen w.)
Relationen: $Int(z,y)$ Meinung z interessiert Person y. (Einschränkung OK)

→ $\neg Int(m(du),ich)$

Brunos Vater ist ein Vorgesetzter von Stefanies Mutter.

Grundmenge: Personen
Konstanten: *br*, *st* Personen
Funktionen: $mu(x)$ Mutter von x
$va(x)$ Vater von x
Relationen: $Vor(x,y)$ x ist ein Vorgesetzter von y.

→ $Vor(va(br), mu(st))$

PL1 und natürliche Sprache – Quantorenmuster (1)

Grundmenge, Konstanten, Funktionen, Prädikate, Relationen?

$\Downarrow$

Formel

- Alle lachen.

  Grundmenge: irgendeine nicht näher bezeichnete Personengruppe
  Prädikate: $L(x)$ – x lacht

  $\forall x\, L(x)$

- (Mindestens) Einer lacht.

  Grundmenge, Prädikat: s.o.

  $\exists x\, L(x)$

- Keiner lacht.

  Grundmenge, Prädikat: s.o.

  $\neg\exists x\, L(x)$

- Alle lachen mich aus (sogar ich mich selbst).

  Grundmenge: s.o.,
  Relation: $La(x,y)$ – x lacht y aus,
  Konstante: i – ich

  $\forall x\, La(x,i)$

PL1 und natürliche Sprache – Quantorenmuster (2)

- Alle (anderen) lachen mich aus.

  Grundmenge, Relation, Konstante: s.o.

  $\forall x\,(x \neq i \rightarrow La(x,i))$

PL1 mit Identität! $x \neq y$ bedeutet $\neg = (x,y)$

- Alle Schadenfrohen lachen mich aus.

  Grundmenge, Relation, Konstante: s.o.
  \+ Prädikat: $Sf(x)$ – x ist schadenfroh

  $\forall x\,(Sf(x) \rightarrow La(x,i))$

Oben sehen wir ein wichtiges Grundmuster:

Für alle mit Eigenschaft *E* gilt … $\forall x\,(E(x) \rightarrow \ldots)$

Üben: Alle Vögel fliegen hoch. (Population: Tiere)
Alle Studierenden lieben Logik. (Population: Menschen)

Etwas schwieriger:
Alle logikfeindlichen Studierenden lieben alle Fächer außer Logik.
Alle Nicht-Studenten bewundern alle Studenten, die Logik verstehen.
(Population: Menschen und Fächer, Prädikate? Relationen?)

PL1 und natürliche Sprache – Quantorenmuster (3)

- Some like it hot.[1] bzw.
  Manche (hier: *mindestens eine(r)*) mögen's heiß. — $\exists x\ Mh(x)$
- Manche Studierenden lieben Logik.

  Grundmenge Menschen (evtl. und Fächer),
  Prädikat: *St* – ist Studierende(r) — **?**

Das andere wichtige Grundmuster:

Für mindestens eines mit Eigenschaft *E* gilt … ➡ $\exists x\ (E(x) \wedge \ldots)$

Also könnte –?– lauten: $\exists x\ (St(x) \wedge liLo(x))$

Oder in der Population aller Menschen <u>und Fächer</u> könnte –?– lauten: $\exists x\ (St(x) \wedge Li(x,lo))$

oder >1 gemeint? Dann …

Üben: Everybody loves somebody.[2]
Alle Studenten verstehen mindestens ein Fach neben Logik.

PL1 mit Identität.

Ü44

---

[1] Billy Wilder
[2] Dean Martin; Blues Brothers

PL1 verallgemeinert AL

... d.h. wir können AL in PL1 **reproduzieren**:

1. durch Beschränkung auf **nullstellige Prädikate**(nsymbole).

   Keine Quantoren, Objekte (Var'en, Konst'en), Funktionen.

   $\boxed{\text{PL1-Theorie liefert AL-Theorie.}}$

2. Sogar allgemeiner, nur:

Keine Quantoren oder Objekt<u>variablen</u>! (Konst'en, Funktionen, zugelassen)

$$(Q(a) \vee \neg R(f(b), c)) \wedge P(a,b) \quad \Longrightarrow \quad (Q\_a \vee \neg R\_f\_b\_c) \wedge P\_a\_b$$

Wie Symbole aufzählen?

PL2 verallgemeinert PL1

In PL2 **quantifiziert** man auch über **Prädikate und Funktionen**, z.B. im **Induktionsaxiom** der natürlichen Zahlen:

Sei P eine mögliche Eigenschaft natürlicher Zahlen, und

- 1 hat die Eigenschaft P, und
- mit jeder Zahl n hat auch $n+1$ die Eigenschaft P.

Dann haben alle natürlichen Zahlen die Eigenschaft P.

$$\boxed{\forall P \langle (P(1) \wedge \forall n [P(n) \rightarrow P(plus(n,1))]) \rightarrow \forall n P(n) \rangle}$$

gebundene Prädikatvariable

aber in axiomatischer Mengenlehre (PL1!) ist beschreib- und beweisbar:

$$\boxed{\forall M \langle (eins \in M \wedge \forall n\, [n \in Nat \cap M \rightarrow plus(n, eins) \in M]) \rightarrow Nat \subseteq M \rangle}$$

PL1-Varianten (1)

Andere **Schreibweisen**

– **Quantoren**:

$\forall x\ F$ auch so: $\bigwedge_{x} F$, $\forall x : F$, $(\forall x)\, F$

$\exists x\ F$ auch so: $\bigvee_{x} F$, $\exists x : F$, $(\exists x)\, F$

– **Relationen**: auch Infix, Mixfix.

– **Relationen- und Funktionensymbole**: $f_1^k, f_2^k, \ldots$, $P_1^k, P_2^k, \ldots$ (pro Stelligkeit k)

– **Relationen- /Funktionensymbole**: 0-stellige erlauben – als Ausdrucksmittel für globale (objektunabhängige) Aussagevariablen bzw. Objektkonstanten (bereits erwähnt)

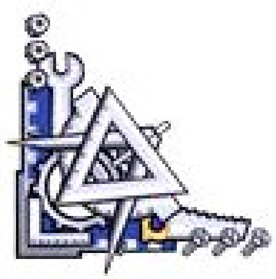
PL1-Varianten (2)

Andere **Formel- und Bedeutungsregeln** (Auswahl)

- PL1 mit **Identität**:
  Vordefinierte zweistellige Relation =. (auch ≡ oder ≈ geschrieben)

- PL1 mit **Sorten**:
  Objekte sind oft **unterschiedlicher** Art, z.B. Zahlen, Vektoren, Matrizen.
  Relationen und Funktionen sind **sortenspezifisch**, z.B. lineare Abbildung ist „Matrix mal Vektor", Kreuzprodukt ist „Vektor mal Vektor".
  Technik: getrennte Variablen- und Konstanten-Namen-Mengen, oder alle Quantoren an Sorten binden: $\forall x \in M\ \exists y \in N\ P(x, y)$.
  Sorten-Ersatz in „unsortierter" PL1: $\forall x(M(x) \rightarrow \exists y\,(N(y) \wedge P(x, y)))$

- PL1 **ohne** besondere **Funktionensymbole**:
  Berechtigung: Funktionen sind spezielle Relationen. Nachteil: Linkstotalität und Rechtseindeutigkeit geeignet explizit „mitzunehmen."



Wichtiger technischer Kleinkram
Freie und gebundene Variablen (1)

Begriff	Definition	Im Formelbaum
[*Text*]	Ein **Vorkommen** (Exemplar) von *Text* in einer Formel. (Kann ja mehrfach vorkommen!)	
Scope, Bereich	In $\forall x\ F$ und $\exists x\ F$ ist ***F*** der Scope vom äußersten $[\forall x]$ /$[\exists x]$.	der ganze Ast unter dem Quantorknoten
Quantorvariable (**-nvorkommen**)	das $[x]$ in $\forall x\ F$ und $\exists x\ F$	

Freie und gebundene Variablen (2)

Begriff	Definition	Im Formelbaum
frei	[*x*] ist frei in *F,* wenn es Teilterm von *F* ist und nicht im Scope eines [∀*x*] / [∃*x*] steht.	
gebunden durch	[*x*] ist gebunden durch [∀*x*] / [∃*x*], wenn es im Scope des [∀*x*] / [∃*x*] frei vorkommt.	Das unterste [∀*x*] / [∃*x*] oberhalb [*x*] bindet [*x*].
geschlossene Formel	... hat **keine freien** Variablen(vorkommen), auch Satz oder Aussage genannt	

also *nicht* Q.-var. in [∀*x*] / [∃*x*].

Theorem?

Aussagenlogik?

Freie und gebundene Variablen im Baum

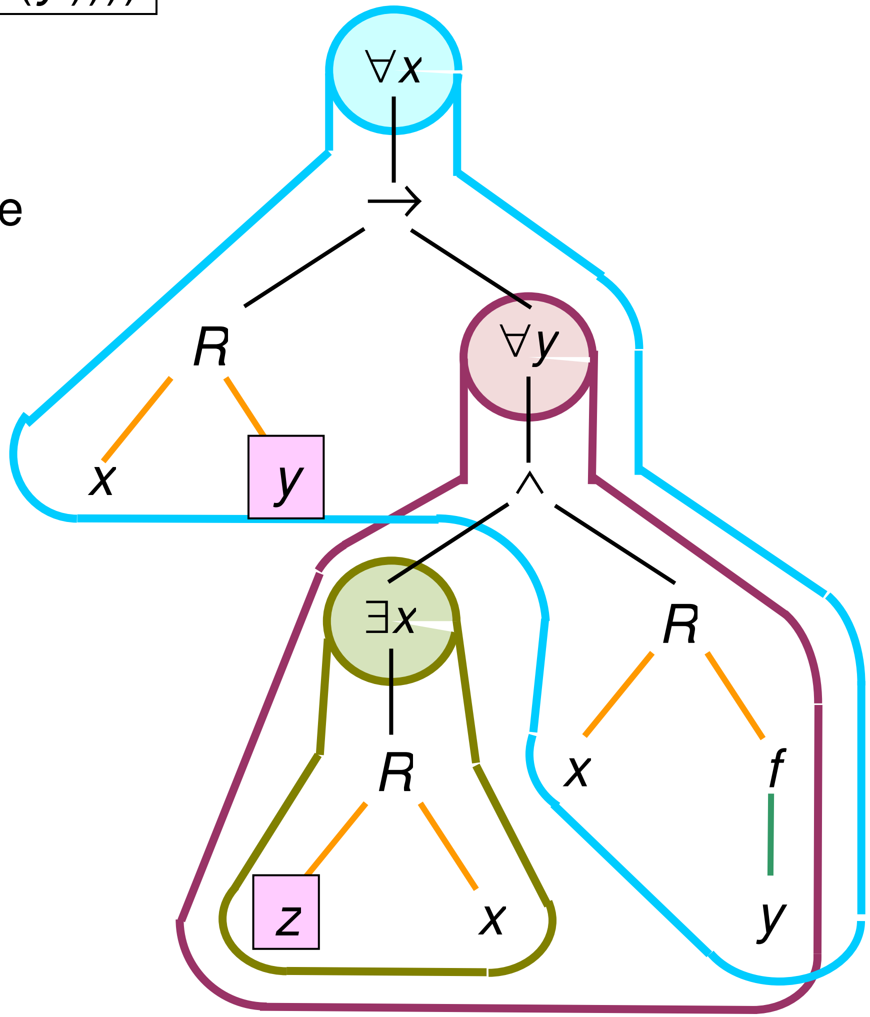
Beispiel : $\boxed{\forall x(R(x,y) \rightarrow \forall y(\exists x R(z,x) \wedge R(x,f(y))))}$

Unterhalb [$\forall x$] sind alle [x] gebunden: Scope!

Allerdings wird die Bindung durch [$\forall x$] teilweise **abgelöst** von der Bindung durch [$\exists x$].

Ein [y] ist frei, ein [y] ist Quantorvariable (das in [$\forall y$]), ein [y] ist gebunden.

Alle [z] sind frei.



Legende

$\boxed{x}$	freies Vorkommen
schwarze Linie	Aufbau komplexer Formel
orange Linie	Aufbau atomarer Formeln
grüne Linie	Aufbau von Termen

Ü45

Prädikatenlogik

- Syntax
- **Semantik**

Interpretationen: Zweistufigkeit

Zuordnungen ...

$$\left.\begin{array}{l}\left.\begin{array}{lll}\text{Konstantensymbole} & \to & \text{Objekte}\\ \text{Funktionssymbole}_k & \to & \text{Funktionen}_k \\ \text{Relationssymbole}_k & \to & \text{Relationen}_k\end{array}\right\}\ \textbf{Struktur}\\ \begin{array}{lll}\text{freie Objektvariablen} & \to & \text{Objekte}\end{array}\qquad\ \ \textbf{Belegung}\end{array}\right\}\ \textbf{Interpretation}$$

Gebundene Variablenvorkommen werden dabei

- „**nicht notwendig interpretiert**“,
- und wenn ja, wird diese **Interpretation** ohnehin **ignoriert**,
- sondern bei der **Auswertung** verwendet und dabei systematisch „**mit allen möglichen Objekten**“ belegt!

Interpretationen: Strukturen

Struktur A=(U,I_A) :

- U = **nicht leere Menge** [1] von Objekten (**Grundmenge**, Universum)
- I_A= **Interpretation der Strukturparameter** – Tripel (I_K,I_F,I_P) von Abbildungen

  $I_K : KS_A \to U,\ KS_A \subseteq KS$
  von (einigen) **Konstantensymbolen** auf Objekte in U;

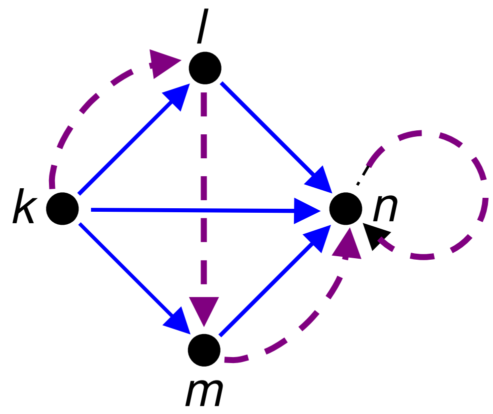
  $I_F : FS_A \to \bigcup_{k=1}^{\infty} F(U^k, U),\ FS_A \subseteq FS$
  von **Funktionssymbolen** auf ein- oder mehrstellige Funktionen auf U, wobei k-stellige Symbole $\mapsto$ k-stellige Funktionen;

  $I_P : PS_A \to \mathbf{P}(\bigcup_{k=0}^{\infty} U^k),\ PS_A \subseteq PS$
  von **Prädikat-** und **Relationssymbolen** auf ein- oder mehrstellige Relationen über U,
  wobei k-stellige Symbole $\mapsto$ k-stellige Relationen.

[1] denkbar: **andersartige** Variante, **LT-Semantik**,
evtl. **l**eerer **T**räger, $\boxed{U = \emptyset}$

$(\forall x\ F) \to (\exists x\ F)$?

Eine Beispiel-Struktur A_Bsp



Alphabet:

Konstantensymbole *KS*: c

Funktionssymbole *FS*: f (1-stellig)

Relationssymbole *RS*: P (2-stellig)

Struktur:

Grundmenge: $U := \{ k, l, m, n \}$

Interpretation:

Konstanten: $I_K : c \mapsto l$

sonst. **Funktionen**: $I_F : f$, gegeben durch $x \dashrightarrow f(x)$

Relationen: $I_P(P)(x, y) := x = y$ oder $x \longrightarrow y$

In einer Struktur können bereits „syntaktisch konstante“ Terme und Formeln ausgewertet werden.

Struktur und Auswertung konstanter Terme

Eine Struktur $A = (U,(I_K, I_F, I_R))$ heißt

passend oder **ausreichend** für PL1-Term τ,

wenn sie **alle Konstanten-, Funktions-, und Prädikatsymbole** in τ **interpretiert**.

In diesem Falle ...

Auswertung von (konstanten) Termen in A (rekursiv):

- c Konstante: $I(c) := I_K(c)$
- f Funktionssymbol und $t_1, t_2, ..., t_k$ Terme:
  $I(f(t_1, ..., t_k))) := I_F(f)(I_A(t_1), ..., I_A(t_k))$

Ü46

Interpretation – Belegung und Auswertung von Termen

Variablen-Interpretation (oder **Belegung**) in einer Struktur $A=(U,I_A)$:

Abbildung $\boxed{I_V : VS_{I_V} \rightarrow U,}$ $VS_{I_V} \subseteq VS$,

von (einigen) Variablen(symbolen) auf Objekte in U.

bzw. (U, I_K, I_F, I_R, I_V)

Ein Paar $\mathcal{I} = (A, I_V)$ = (Struktur, Belegung), eine **Interpretation**, heißt **passend** oder **ausreichend** für eine PL1-Formel φ, wenn es alle

- Konstanten-, Funktions-, und Prädikatsymbole,
- sowie alle **freien** Variablen in φ interpretiert.

In diesem Falle ist A ausreichend für alle Terme in φ ohne gebundene Variablen, und die **Formel** φ kann **ausgewertet** werden.

Auswertung von „beliebigen" Termen: $I(\tau)$ wie $I_A(\tau)$ + zusätzlich ...

- x Variable (genügt: [x] freies Variablenvorkommen): $I(x) := I_V(x)$

Belegung und Formel-Auswertung (1)

Auswertung von **Formeln** unter passendem I (rekursiv):

- **atomare Formeln**: I sei ausreichend für $P(t_1, ..., t_k)$, *wobei* P Prädikatsymbol und $t_1, t_2, ..., t_k$ Terme:

  $$I\big(P(t_1, ..., t_k)\big) := \begin{cases} W & \text{wenn} \quad \big(I(t_1), ..., I(t_k)\big) \in I_P(P) \\ F & \text{sonst} \end{cases}$$

- **„Junktorenformeln"**: I sei ausreichend für Formeln φ, ψ.

  $I(\neg\varphi), I(\varphi \wedge \psi), I(\varphi \vee \psi), I(\varphi \rightarrow \psi), I(\varphi \leftrightarrow \psi)$ ergeben sich aus $I(\varphi), I(\psi)$ wie in der **Aussagenlogik**.

Belegung und Formel-Auswertung (2)

„Quantorenformeln“: I sei ausreichend für Formel $\forall x \varphi$

- $I(\forall x \varphi) := \begin{cases} \text{W} & \text{wenn für alle } a \in U \text{ gilt}: I_{x:=a}(\varphi) = \text{W} \\ \text{F} & \text{sonst} \end{cases}$

Belegung außerhalb x festhalten. Mit x das ganze U durchprüfen.

- $I(\exists x \varphi) := \begin{cases} \text{W} & \text{wenn es } a \in U \text{ gibt mit}: I_{x:=a}(\varphi) = \text{W} \\ \text{F} & \text{sonst} \end{cases}$

Wie I,
aber $I_V(x) := a$

Unterschied $x{:=}a$ / $[x/a]$? :
$x{:=}a$ wertemäßig, nicht textlich; auch für „namenlose“ a; betr. nur freie Vorkommen!

Ü47

Die zwei Richtungen der PL1-Formel-Auswertung

Die Interpretation und die Auswertung

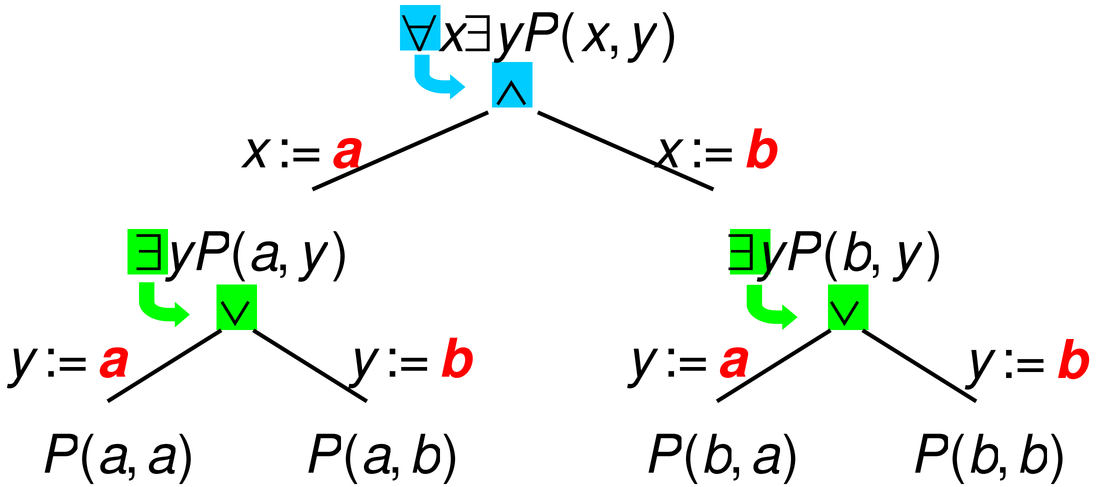
- verlaufen also in jeder quantorfreien Formel von innen nach außen
  (im Formelbaum: von den Blättern zur Wurzel).

Für jede quantisierte Teilformel

- werden aber von außen nach innen
  (im Formelbaum: von der Wurzel weg zu den Blättern)
  Auswertungen für im Prinzip alle Belegungen der quantifizierten Variablen
  mit Werten aus ganz U angestoßen,
  was zu einer Vielzahl (meist unendlichen Zahl)
  von Auswertungen quantorfreier Formeln führt.

Ein Beispiel für die PL1-Formel-Auswertung

Sei **U={ *a*,*b* }**.



… entspricht also:

$$[\,(\;P(a,a)\;\vee\;P(a,b)\;)\;\wedge\;(\;P(b,a)\;\vee\;P(b,b)\;)\,]$$

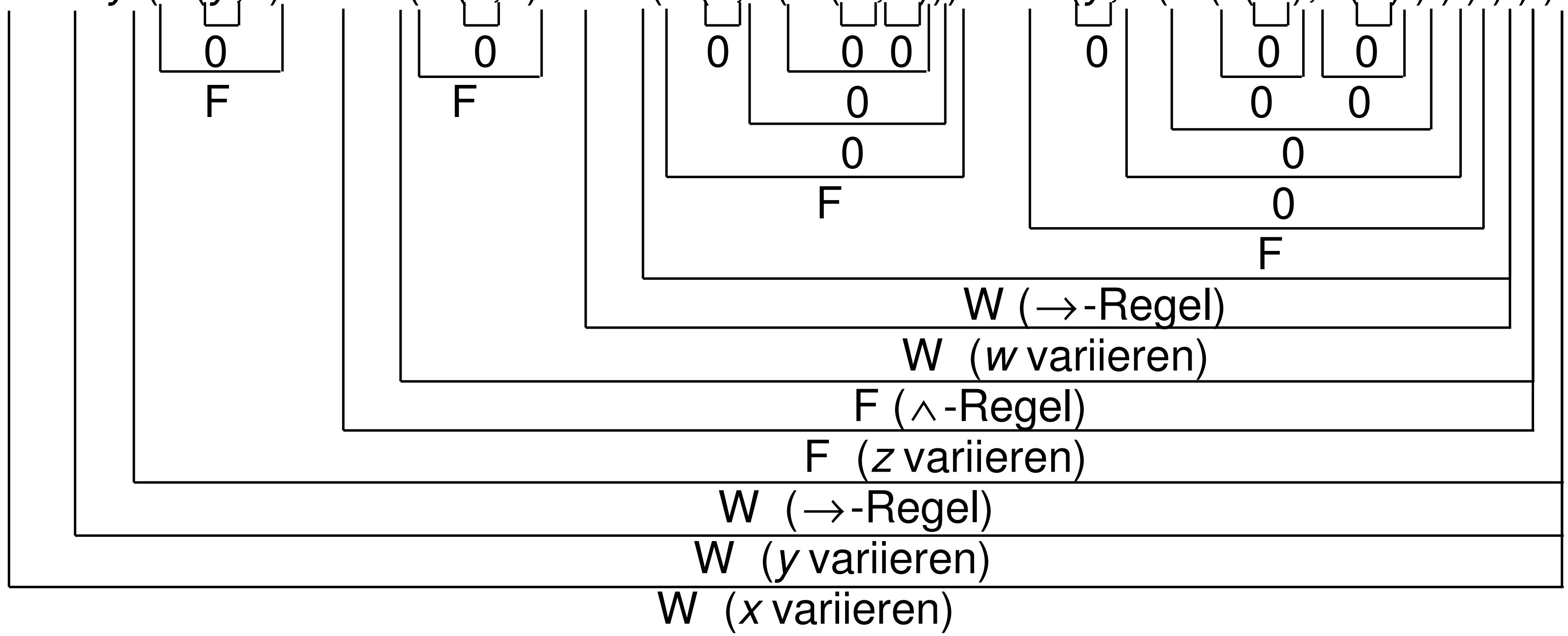
Formel-Auswertung unter verschiedenen Interpretationen (1)

Formel $\boxed{\forall x \forall y (G(y,n) \rightarrow \exists z (G(z,n) \wedge \forall w (G(z, b(m(w,x))) \rightarrow G(y, b(m(f(w), f(x)))))))}$ mit ...

- Objektkonstante n
- Objektvariablen w,x,y,z
- Funktionssymbolen m (2-stellig), b,f (1-stellig)
- Relationssymbolen G (2-stellig)

1. Interpretation $U = \{0\}$, $n,w,x,y,z \mapsto 0$, b,f,m auch klar (wieso?), $G \mapsto \{\,\}$

$\forall x\ \forall y\ (G(y,0) \rightarrow \exists z\ (G(z,0) \wedge \forall w (G(z, b(m(w,x))) \rightarrow G(y, b(m(f(w), f(x)))))))$



Formel-Auswertung unter verschiedenen Interpretationen (2)

Formel $\boxed{\forall x \forall y (G(y,n) \rightarrow \exists z (G(z,n) \wedge \forall w (G(z,b(m(w,x))) \rightarrow G(y,b(m(f(w),f(x)))))))}$ mit ...

- Objektkonstante *n*
- Objektvariablen *w,x,y,z*
- Funktionssymbolen *m* (2-stellig), *b*,*f* (1-stellig)
- Relationssymbolen *G* (2-stellig)

2. Interpretation

$U = I\!R,\; n, \underbrace{w, x, y, z}_{\text{fiktiv}} \mapsto 0,\; , b(x) := |x|\,,\; f(x) := x^2\,,\; m(x,y) := x - y, G(x,y) :\Leftrightarrow x > y$

Etwas anders geschreiben:

$\forall x_0 \forall \varepsilon > 0 \; \exists \delta > 0 : \forall x (| x - x_0 | < \delta \rightarrow | x^2 - x_0{}^2 | < \varepsilon)))$

$\Rightarrow$ $f(x) = x^2$ stetig?

Wahr!

Ü48

Prädikatenlogik

- Syntax
- Semantik
- **Formeln, Modelle, Tautologien und Anwendungen**

Semantische Begriffe (1)

Vergleichen Sie mit der Aussagenlogik!

Eine zu einer Formel φ passende Interpretation $I = (A, I_V)$ heißt

Modell von φ $:\Leftrightarrow$ $I(\varphi) = \mathrm{W}$

Schreibweise: $\boxed{I \models \varphi}$

Redeweise: φ **gilt unter *I***, bzw. ***I* erfüllt** φ.

Die Formel φ heißt

- **erfüllbar**, wenn sie mindestens **ein** Modell hat (sonst: **unerfüllbar**);
- **allgemeingültig** (logisch wahr, $\boxed{\models \varphi}$)
  wenn **jede** passende Interpretation Modell ist (sonst: **widerlegbar**);
- **gültig in** der Struktur ***A*** ($\boxed{A \models \varphi}$),
  wenn für jede passende Belegung I_V die Interpretation $I = (A, I_V)$ Modell ist.

  Beispiel: $a + b = b + a$ „gilt in $\mathbb{N}_0$."

Semantische Begriffe (2)

Zwei Formeln φ und ψ heißen

- (logisch, semantisch) **äquivalent**, $\boxed{\varphi \equiv \psi}$,
  wenn unter jeder passenden Interpretation I gilt: $I(\varphi) = I(\psi)$.

Eine Formel φ

- **folgt** (logisch, semantisch) aus einer Formel σ (Formelmenge *M*),
  geschrieben: $\boxed{\sigma \models \varphi}$ ($\boxed{M \models \varphi}$),
  wenn jedes Modell von σ (bzw. *M*) auch Modell von φ ist.

Wichtige Äquivalenzen (1)

Erinnerung:

> Zwei Formeln φ und ψ heißen (logisch, semantisch) **äquivalent**, $\varphi \equiv \psi$, wenn unter jeder zu beiden passenden Interpretation I gilt: $I(\varphi) = I(\psi)$.

Dualität von $\forall$ und $\exists$

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

Vertauschung der Quantoren-Reihenfolge

nur $\forall/\forall$ oder $\exists/\exists$
– nicht $\forall/\exists$!

- $\forall x \forall y\, \varphi \equiv \forall y \forall x\, \varphi$
- $\exists x \exists y\, \varphi \equiv \exists y \exists x\, \varphi$

Quantoreinführung und **-beseitigung** – Voraussetzung: x **nicht frei** in φ

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

Ü49

Wichtige Äquivalenzen (2)

(Leere) **Scope-Erweiterung** (auf ψ) – Voraussetzung: x **nicht frei** in ψ.

- $\forall x\, \varphi \wedge \psi \equiv \forall x\, (\varphi \wedge \psi)$
- $\forall x\, \varphi \vee \psi \equiv \forall x\, (\varphi \vee \psi)$
- $\psi \rightarrow \forall x \varphi \equiv \forall x (\psi \rightarrow \varphi)$
- $\exists x\, \varphi \vee \psi \equiv \exists x\, (\varphi \vee \psi)$
- $\exists x\, \varphi \wedge \psi \equiv \exists x\, (\varphi \wedge \psi)$
- $\psi \rightarrow \exists x \varphi \equiv \exists x (\psi \rightarrow \varphi)$

aber Achtung: $\forall x \varphi \rightarrow \psi \not\equiv \forall x\, (\varphi \rightarrow \psi)$!

Distributivität

- $\forall x\, \varphi \wedge \forall x \psi \equiv \forall x\, (\varphi \wedge \psi)$
- $\exists x\, \varphi \vee \exists x \psi \equiv \exists x\, (\varphi \vee \psi)$

Nicht „über Kreuz", d.h. $\forall + \vee$ / $\exists + \wedge$!

Gegenbeispiele in $\mathbb{N}_0$?

Z.B. mit x>0, 0>0, x un/gerade, x=0/1

„AL-Substitution“ in PL1

Satz über AL-Tautologien in PL1:

Werden in einer AL-Tautologie (bzw. in einer unerfüllbaren AL-Formel)
alle Vorkommen jeder Aussagevariablen
jeweils durch die gleiche PL1-Formel ersetzt,
so ist die entstehende PL1-Formel ebenfalls eine Tautologie (bzw. unerfüllbar).

Dies ist eine Abart der **AL-Substitution**, formale Definition wie dort!

Ersetzungssatz in PL1

Ersetzungssatz (Satz über die äquivalente Ersetzung):

Werden in einer Formel φ
ein oder mehrere Vorkommen einer Teilformel ψ
durch eine zu ψ äquivalente Formel ρ ersetzt,
so ist die entstehende Formel zu φ äquivalent.

Die Begründung ist die gleiche wie bei der **AL-Ersetzung**!

PL1-Substitution (1)

Die **Substitution in PL1** $\Longrightarrow$ $\varphi \mapsto \varphi_{[x_1, \ldots, x_n / \tau_1, \ldots, \tau_n]}$

ersetzt keine Aussagen-, sondern **Objekt**variablen, nämlich:
paarweise verschiedene Objektvariablen $x_1, \ldots, x_n$
durch (nicht unbedingt verschiedene) Terme $\tau_1, \ldots, \tau_n$.

Termebene:

$x \in OV \cup OK$: $\quad x_{\overbrace{[x_1, \ldots, x_n / \tau_1, \ldots, \tau_n]}^{sub}} := \begin{cases} \tau_i & \text{für } x = x_i \\ x & \text{sonst} \end{cases}$,

$f \in FS \quad f(\omega_1, \ldots, \omega_k)_{[sub]} := f(\omega_{1[sub]}, \ldots, \omega_{k[sub]})$

Übergang zur Aussagenebene:

$R \in RS \quad R(\omega_1, \ldots, \omega_k)_{[sub]} := R(\omega_{1[sub]}, \ldots, \omega_{k[sub]})$

wie zu
erwarten

PL2-Substitution (2) und gebundene Umbenennung

Aussagenebene:

Negation $(\neg\varphi)_{[sub]} := \neg(\varphi_{[sub]})$

$\otimes \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$: $(\varphi \otimes \rho)_{[sub]} := \varphi_{[sub]} \otimes \rho_{[sub]}$

wie zu erwarten

Quantoren $(\forall x\, \varphi)_{[sub]} := \forall x(\varphi_{[sub \neq x]})$

$(\exists x\, \varphi)_{[sub]} := \exists x(\varphi_{[sub \neq x]})$

$sub_{\neq x}$ ist „wie *sub*, aber das gebundene x wird nicht ersetzt“.

!

Warum?

Satz über die gebundene Umbenennung
Sei φ eine Formel, die die Variable y nicht frei enthält.
Dann ist $\exists x\, \varphi \equiv \exists y\, \varphi_{[x/y]}$ und
$\forall x\, \varphi \equiv \forall y\, \varphi_{[x/y]}$.

Nicht-Substituierbarkeit

Was im Allgemeinen gilt,
gilt doch auch im Besonderen ...
Also gilt doch für jede Substitution:
$\forall x\varphi \to \varphi_{[x/\tau]}$?

NEIN!

Wie kann das schief gehen?

z.B. so, mit $[x/y]$:

$\forall x(\neg\forall y\ x = y) \to \neg\forall y\ y = y$ **??**

$\forall x(\exists y\ x < y) \to \exists y\ y < y$ **??**

Aha, also wenn dadurch in φ
„ein freies Variablenvorkommen
aus τ **gebunden** wird!"

Wie formalisieren wir das?

(Nicht-)Substituierbarkeit & Substitutionssatz

Term τ ist **<u>nicht</u> erlaubt (nicht frei, nicht substituierbar) für *x* in** φ ,
$\neg$free(τ , x, φ),

:$\Leftrightarrow$ Es existieren ...

- Variablenvorkommen [y] in τ,
- ein freies Variablenvorkommen [x] in φ , $x \neq y$
- ein Quantorenvorkommen [Qy] in φ , so dass
- dieses [x] im Scope von [Qy] liegt

$[x \,/\, \underbrace{y}_{\tau}]$

$\underbrace{\forall x(\neg\forall y \; x = y)}_{\varphi}$

Substitutionssatz

free(τ , x, φ) $\Rightarrow$ $\forall x \, \varphi \rightarrow \varphi_{[x/\tau]}$

Substituierbar (d.h. τ frei für x_i in φ)? – ein Algorithmus

Geg.: τ, x_i und φ (i ist fest)

1) Bestimme alle freien [x_i] (freie Vorkommen des gegebenen x_i) in φ.

   Gibt es keine ➔ τ ist **frei** für x_i in φ & <u>Stop</u>.

2) Für alle freien [x_i] in φ:

   Bestimme in φ alle Quantisierungen [Qx_k], automatisch mit $i \neq k$, in deren Scope dieses [x_i] liegt.

   (i) Gibt es keine ➔ τ ist **frei** für x_i in φ & <u>Stop</u>.

   (ii) Für alle diese [Qx_k]:
   Kommt x_k in τ vor ➔ τ ist **nicht frei** für x_i in φ & <u>Stop</u>.

3) [*Sind nun also alle freien Vorkommen* [x_i] *in* φ *durchprobiert, ohne auf <u>Stop</u> zu laufen* ➔]

   τ ist **frei** für x_i in φ & <u>Stop</u>.

Ü50

Manche sagen auch „x frei für τ“ anstatt „τ frei für x“.

Einfache Folgerungen

- τ enthält keine Variable ➔ τ frei für alle Variablen in allen Formeln.
- keine Variable in τ gebunden in φ ➔ τ frei für alle Variablen in φ.
- stets: x_i ist frei für x_i in (jedem) φ .
- φ enthält kein freies [x_i] ➔ (jedes) τ ist frei für x_i in φ .

Prädikatenlogik

- Syntax
- Semantik
- Formeln, Modelle, Tautologien und Anwendungen
- **Entscheidbarkeitsfragen**

PL1-Vorteil: Fast die gesamte Mathematik lässt sich damit ausdrücken.

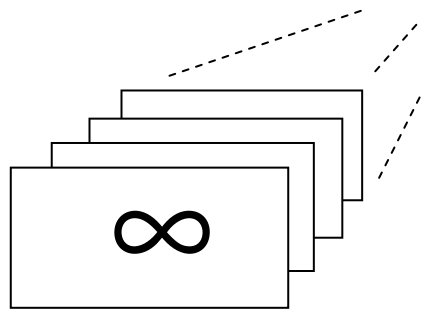
In der **Aussagenlogik** sind **semantische Eigenschaften** – bis evtl. auf Folgerung aus unendlichen Formelmengen – prinzipiell **entscheidbar:** die ...

Wahrheitstafel

... erlaubt die vollständige Prüfung anhand aller (endlich vielen!) **Interpretationen** (= AL-Belegungen).

PL1-Nachteil

Jetzt, in der Prädikatenlogik, wären
unendlich viele Strukturen
und pro unendlicher Struktur jeweils
unendlich viele Variablen-Interpretationen
zu untersuchen.



(Un-)Entscheidbarkeit (1)

Es gibt keinen Algorithmus, der

- Erfüllbarkeit,
- Allgemeingültigkeit oder
- Folgerung

entscheidet.

aber: **Beweise** können algorithmisch **überprüft** werden!

Nie – oder nur **noch keinen?**

… weil lediglich noch keiner schlau genug war einen zu (er)finden?

Nein: „Vermutlich endgültig“, weil es „**fast** beweisbar“ **keinen geben kann**! (Church, Turing 1936)

Wie zeigt man so etwas?

Und wieso „fast“?

(Un-)Entscheidbarkeit (2): Churchsche These

Mathematische Definition des Begriffs des (beliebigen!) **Algorithmus**, z.B.

- **Turing-Maschine** (TM): elementare Programmschritte + Speicher
- andere formalisierte Algorithmusbegriffe
  (Rekursive Funktionen, Markov-Systeme, Post-Systeme, ...)

Erfahrung (bewiesen!):
Alle bisherigen Algorithmus-Klassen berechnen die gleiche Klasse von Funktionen.

Churchsche These:
Die Klasse der intuitiv berechenbaren Funktionen ist genau die Klasse der Turing-berechenbaren Funktionen.

Was also mit Turing-Maschinen nicht berechenbar ist,
dafür gibt es – laut Churchscher These – überhaupt keinen Algorithmus.

Also reden wir jetzt mal nur noch über Turing-Maschinen.

(Un-)Entscheidbarkeit (3): Halteproblem

Technische Überlegungen →
Wenn eine TM über Erfüllbarkeit von PL1-Formeln entscheiden könne, ließe sie sich umformen zu einer einer TM, die entscheidet, ob eine gegebene TM auf einem gegebenen Input mit einem Ergebnis stoppt.

Halteproblem
Es gibt aber keine TM, die für jede TM und beliebige Eingaben entscheidet, ob diese TM auf dieser Eingabe terminiert!

Beweisskizze

Annahme, es ginge **doch**, sagen wir mit der TM T_{super} .

Zähle **alle** Turingmaschinen durch: T_1, T_2, T_3, T_4, … (**geht**: Gödel-Nummern!).

Definiere neue TM $T_{\mathbf{seltsam}}$ durch

$$T_{seltsam}(T_i) := \begin{cases} T_i(T_i)+1 & \textit{wenn } T_i(T_i) \textit{ existiert} \\ 0 & \textit{wenn } T_i(T_i) \textit{ nicht stoppt} \end{cases}$$

Wegen T_{super} berechenbar!!

Diagonalargument: Dann kann $T_{\mathbf{seltsam}}$ nicht eine der T_i sein.

→ ein **Widerspruch;** die **Annahme** muss **falsch** sein.

Halbentscheidungs-Algorithmen

Eine Frage(nmenge) ist durch Algorithmus *Alg* **halbentscheidbar** :⇔

- Falls korrekte Antwort JA ⇒ *Alg* endet irgendwann mit JA;
- Falls NEIN: *Alg* endet nicht

wieso?

Für die **Allgemeingültigkeit** (**also** auch **Unerfüllbarkeit** und **Folgerung**) lernen wir noch
Beweisregeln und **Halbentscheidungs-Algorithmen** kennen:

- Beweiskalküle
- Beweisalgorithmus
- PL1-Tableaux
- PL1-Resolution
- Natürliches Schließen (Werkzeugkasten)

wieso?

Also kann es **keinen Halbentscheidungs-Algorithmus** für die **Erfüllbarkeit** (**Widerlegbarkeit**, **Nicht-Folgerung**) geben!

Und wenn ich das Gegenteil halb-
entscheidbar auf Unerfüllbarkeit prüfe??

Anwendung pfiffiger Beweismethoden
(1) Diagonalverfahren

Was haben

- der Beweisgedanke der **Unentscheidbarkeit des Halteproblems** und
- der Standardbeweis für die **Überabzählbarkeit der reellen Zahlen**

gemeinsam?

Zu einer unendlichen Matrix mit Zeilen z_i wird per geschickter* **Änderung aller Diagonalemente** eine „neue Zeile“ generiert,

$$\begin{array}{lcccc} z_1: & \boxed{a_{11}} & a_{12} & a_{13} & \cdots \\ z_2: & a_{21} & \boxed{a_{22}} & a_{23} & \cdots \\ z_3: & a_{31} & a_{32} & \boxed{a_{33}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

die sich von der i-ten Matrixzeile z_i an der i-ten Stelle unterscheidet und daher **nicht** in der Aufzählung $z_1, z_2, z_3, \ldots$ **vorkommen kann**.

Beim **Halteproblem** ist a_{ik} das Ergebnis der Anwendung von Maschine T_i auf das Programm der Maschine T_k (ggf. *undefiniert* wegen Nichtterminierung – die angenommene T_{super} sagt uns wann das der Fall ist!). Bei den **rellen Zahlen** (sagen wir: zwischen 0 und 1) ist a_{ik} die k-te Dezimalstelle hinter dem Komma.

*) In beiden Fällen muss eine **legitime Zeile** erzeugt werden!

Anwendung pfiffiger Beweismethoden (2) Koordinierte Halbentscheidungsalgorithmen

Wieso kann es keinen Halbentscheidungsalgorithmus für die Erfüllbarkeit von PL1-Formeln geben?

- **Angenommen es gäbe einen: ERF.**
- GRES (Grundresolution) ist ein Halbentscheidungsalgorithmus für Unerfüllbarkeit – den lernen wir noch kennen.
- Nun programmiert man leicht einen übergeordneten Algorithmus ÜA, der abwechselnd einen „Schritt" von ERF und einen von GRES durchführt und wendet ihn jeweils auf eine (evtl. geeignet umgeformte) PL1-Formel φ an. ÜA soll stoppen sobald er mit ERF oder GRES auf ein *Stop* läuft.
- φ ist entw. erfüllbar oder unerfüllbar, d.h. ERF oder GRES läuft in endlich vielen Schritten auf ein (bejahendes!) *Stop*, sagen wir: nach n Schritten.
- D.h. ÜA wäre ein Algorithmus, der (bei φ in max. $2n$ Schritten) entscheidet ob φ erfüllbar ist. Damit wäre das Halteproblem lösbar – Widerspruch!
- Also gibt es dieses ÜA nicht, also muss die Annahme falsch sein.
- **Dieses ERF gibt es nicht. (Widerspruchsbeweis)**

Prädikatenlogik

- Syntax
- Semantik
- Formeln, Modelle, Tautologien und Anwendungen
- Entscheidbarkeitsfragen
- **Kalküle und Tableaux**

Ein(e Gruppe von) Prädikatenkalkül(en) (1)

PL1-Formeln

- Bekannt (mit Aussagevariablen)

Regeln

- Modus Ponens $\boxed{\mathrm{MP} = \dfrac{\varphi, \varphi \to \psi}{\psi}}$

Ein(e Gruppe von) Prädikatenkalkül(en) (2)

Axiome(nschemata) (für bel. Formel φ und bel. Variable x)

- ein AL-Axiomensystem — Aussagenlogik AL

und <u>beliebige Verallgemeinerungen</u> $\forall x_1 \ldots \forall x_n$ von Formeln der Arten ...

- $\varphi_{[A_1,\ldots,A_k \,/\, \varphi_1,\ldots,\varphi_k]}$ wobei φ AL-Tautologie, φ_i Formeln (AL-Gültigkeit AL-G)
- $(\forall x\varphi) \to \varphi_{[x\,/\,\tau]}$ sofern τ Term, der für x in φ erlaubt ist: free(τ,x,φ) (Substitutionssatz, Spezialisierung SP)
- $\varphi \to (\forall x\varphi)$ sofern x nicht frei in φ vorkommt (Generalisierung GN)
- $\forall x(\varphi \to \psi) \to (\forall x\varphi \to \forall x\psi)$ (Generalisierungsdistribution GD)
- $\exists x\varphi \leftrightarrow \neg\forall x\neg\varphi$ (Existenz„definition“ ED)

Anwendung des Prädikatenkalküls (1)

Korrektheit dieser PL1-Kalküle

Alle ableitbaren Formeln sind allgemeingültig.

Alle aus einer Formelmenge *M* ableitbaren Formeln sind Folgerungen von *M*.

Gödelscher Vollständigkeitssatz:

... und umgekehrt! D.h. alle allgemeingültigen Formeln und Folgerungen sind damit ableitbar.

warum fordern?

Generalisierungssatz **GS**

$M \vdash \varphi$ und x in keinem $\psi \in M$ frei $\Rightarrow$ $M \vdash \forall x \varphi$

Deduktionssatz: **DS**

$M \cup \{\varphi\} \vdash \psi \Rightarrow M \vdash \varphi \rightarrow \psi$

Anwendung des Prädikatenkalküls (2)

Beispiel für eine Meta-Ableitung (d.h. unter Verwendung semantischer Sätze)

1.	$(A_1 \to \neg A_2) \to (A_2 \to \neg A_1)$	aus AL, MP (mehrfach)
2.	$(\forall y \neg P(y) \to \neg P(x)) \to (P(x) \to \neg\forall y \neg P(y))$	1, AL-G
3.	$\forall x\,[(\forall y \neg P(y) \to \neg P(x)) \to (P(x) \to \neg\forall y \neg P(y))]$	2, GS (M=∅)
4.	$\forall x\,(\forall y \neg P(y) \to \neg P(x)) \to \forall x(P(x) \to \neg\forall y \neg P(y))$	3, GD
5.	$\forall y \neg P(y) \to \neg P(x)$	SP
6.	$\forall x\,(\forall y \neg P(y) \to \neg P(x))$	5, GS
7.	$\forall x(P(x) \to \neg\forall y \neg P(y))$	4, 6, MP

Meta-Ableitungen garantieren die Existenz von – meist deutlich längeren – Ableitungen (bzw. liefern sie).

Halbentscheidung mit „roher Gewalt“

Wir wollen φ auf **Allgemeingültigkeit** prüfen.

Dazu **produzieren** wir mittels eines PL1-Kalküls
systematisch* der Reihe nach
alle möglichen Ableitungen.

Nach jeder solchen Produktion:
Erzeugte **Ableitung endet mit φ?**
ja: JA, Stop.
nein: nächste Produktion.

Man kann aber auch gezielter vorgehen:

PL1-Algorithmen (Tableaux, Resolution etc.)

---

*) **Wie geht das?** SP hat doch ∞ viele Varianten!
(So nicht: Stufe *n*: *n* Regel-Anwendungen auf alle Axiome. n=0, 1, ...**)**

PL1-Tableaux (1)

Anwendung für **geschlossene Formeln** (ohne freie Variablen)

Zusätzliche Regeln (zu den aussagenlogischen Tableauregeln)

Speziali-sierung	$\mathrm{Sp}_+ = \dfrac{\forall x \varphi}{\varphi_{[x/\tau]}}$	$\mathrm{Sp}_- = \dfrac{\neg\exists x \varphi}{\neg\varphi_{[x/\tau]}}$	τ für x in φ zulässig, d.h. free(τ, x, φ)
Witness (Zeuge)	$\mathrm{W}_+ = \dfrac{\exists x \varphi}{\varphi_{[x/c]}}$	$\mathrm{W}_- = \dfrac{\neg\forall x \varphi}{\neg\varphi_{[x/c]}}$	c Konstanten-symbol, das im Zweig neu ist

PL1-Tableaux – Anwendungsbeispiel

Zu beweisen: $\forall x(P(x) \vee Q(x)) \rightarrow (\exists x P(x) \vee \forall x Q(x))$

Gegenbeh.:		Regel / Prämisse	
1.	$\neg[\forall x(P(x) \vee Q(x)) \rightarrow (\exists x P(x) \vee \forall x Q(x))]$		
2.	$\forall x(P(x) \vee Q(x))$	(TR1)	1
3.	$\neg(\exists x P(x) \vee \forall x Q(x))$	(TR1)	1
4.	$\neg\exists x P(x)$	(TR3)	3
5.	$\neg\forall x Q(x)$	(TR3)	3
6.	$\neg Q(c)$ ↯	W_-	5
7.	↯ $\neg P(c)$	Sp_-	4
8.	$P(c) \vee Q(c)$	Sp_+	2
9. / 10.	$\underline{P(c)}$ | $\underline{Q(c)}$	(TR8)	8

Tableau-regeln AL, Erinnerung

1. $\dfrac{\neg(\varphi \rightarrow \psi)}{\varphi \quad \neg\psi}$

3. $\dfrac{\neg(\varphi \vee \psi)}{\neg\varphi \quad \neg\psi}$

8. $\dfrac{\varphi \vee \psi}{\varphi \mid \psi}$

Ü51

PL1-Tableaux-Eigenschaften

Als **Kalkül** ist PL1-Tableaux **korrekt** und **vollständig**:

- Enthalten alle Zweige einen Widerspruch zwischen zwei Literalen, ist die Wurzel unerfüllbar bzw. ihre Negation allgemeingültig
- Ist die Wurzel unerfüllbar bzw. ihre Negation allgemeingültig, so existieren abgeleitete Tableaux, die in allen Zweigen einen Widerspruch zwischen zwei Literalen enthalten.

Als **Algorithmus** ist PL1-Tableaux leider **nicht** unbedingt **terminierend.**

Bei geeigneter Variation und Steuerung, z.B.

- im Rahmen eines Brute-Force-Verfahrens oder
- (weitaus effizienter) mit raffinierteren Techniken

taugen sie als **Halbentscheidungsverfahren**:

Die Terminierung bei unerfüllbarer Wurzel kann garantiert werden.

PL1-Werkzeug-Ergänzungs-Kasten (1)

All-Benutzung Spezialisierung	Sp $\dfrac{\forall x\varphi}{\varphi_{[x/\tau]}}$ (τ variablenfrei)	Existenz-Einführung	EE $\dfrac{\varphi_{[x/\tau]}}{\exists x\varphi}$ (τ variablenfrei)
Existenz-Benutzung	EB $\dfrac{\exists x\varphi}{\varphi_{[x/c]}}$ (c neu, vgl. All-Beweis)[3]		
Nicht-Existenz-Benutzung	NEB $\dfrac{\neg\exists x\varphi}{\forall x\neg\varphi}$	Nicht-All-Benutzung	NAB $\dfrac{\neg\forall x\varphi}{\exists x\neg\varphi}$

All-Beweis-Schema AB: Zeige $\forall x\varphi$, unmittelbar gefolgt von weiterem Blockbeginn: Zeige $\varphi_{[x/c]}$ (*c* im Beweis **neuer Konstantenname**) + Blockbeginn. Sobald im gleichen Block $\varphi_{[x/c]}$ erreicht: Blockende. Dann weiteres Blockende mit $\forall x\varphi$ (AB).

Zeige $\forall x\varphi$
| Zeige $\varphi_{[x/c]}$
| | ⋮
| $\varphi_{[x/c]}$ (...B)
$\forall x\varphi$ (AB)

[3] Kein Beweis der Formel mit *c*, nur eine temporäre Namensgebung: „Nennen wir es c!“!

PL1-Werkzeug-Ergänzungs-Kasten (2)

Der AL&PL1-Werkzeugkasten
ist **korrekt** und **vollständig**
für **geschlossene** PL1-Formeln.

2 Beweisstrategien für Existenzaussagen:

Direkter Beweis:

Zeige $\exists x \varphi$
| $\vdots$
| $\varphi_{[x/\tau]}$
| $\exists x \varphi$ (EE)
$\exists x \varphi$ (DB)

Indirekter Beweis:

Zeige $\exists x \varphi$
| $\neg \exists x \varphi$ (Ann)
| $\vdots$
| $\perp$ (WE)
$\exists x \varphi$ (IB)

Beispiel 1 für die Arbeit mit dem kompletten Werkzeugkasten

Beweisen Sie die Korrektheit von

$$\forall x P(x) \rightarrow \exists y P(y)$$

	Zeige $\forall x P(x) \rightarrow \exists y P(y)$	
(1)	$\mid$ $\forall x P(x)$	Ann.
	$\mid$ Zeige $\exists y P(y)$	
(2)	$\mid$ $\mid$ $P(a)$	1, Sp
(3)	$\mid$ $\mid$ $\exists y P(y)$	2, EE
(4)	$\mid$ $\exists y P(y)$	DB
(5)	$\forall x P(x) \rightarrow \exists y P(y)$	BB

Beispiel 2 für die Arbeit mit dem kompletten Werkzeugkasten

Beweisen Sie die Korrektheit der Folgerung

$$\{\forall x(P(x) \rightarrow Q(x)), \neg Q(a)\} \models \neg \forall y P(y)$$

(1)	$\forall x(P(x) \rightarrow Q(x))$		Geg.
(2)	$\neg Q(a)$		Geg.
	Zeige $\neg \forall y P(y)$		
(3)	|	$\forall y P(y)$	Ann.
(4)	|	$P(a)$	3, Sp
(5)	|	$P(a) \rightarrow Q(a)$	1, Sp
(6)	|	$Q(a)$	4, 5, MP
(7)	|	$\perp$	2, 6, WE
(8)	$\neg \forall y P(y)$		IB

Beispiel 3 für die Arbeit mit dem kompletten Werkzeugkasten

Beweisen Sie

$$\forall x(\neg P(x) \to \neg\forall y P(y))$$

$$
\begin{array}{lll}
 & \text{Zeige } \forall x(\neg P(x) \to \neg\forall y P(y)) & \\
 & | \quad \text{Zeige } \neg P(c) \to \neg\forall y P(y) & \\
(1) & | \quad | \quad \neg P(c) & \text{Ann.} \\
 & | \quad | \quad \text{Zeige } \neg\forall y P(y) & \\
(2) & | \quad | \quad | \quad \forall y P(y) & \text{Ann.} \\
(3) & | \quad | \quad | \quad P(c) & \text{2, Sp} \\
(4) & | \quad | \quad | \quad \bot & \text{1,3, WE} \\
(5) & | \quad | \quad \neg\forall y P(y) & \text{IB} \\
(6) & \quad \neg P(c) \to \neg\forall y P(y) & \text{BB} \\
(7) & \forall x(\neg P(x) \to \neg\forall y P(y)) & \text{AB}
\end{array}
$$

Ü52