THEORETISCHE INFORMATIK UND LOGIK

19. Vorlesung: Resolution

Markus Krötzsch

Lehrstuhl Wissensbasierte Systeme

TU Dresden, 28. Juni 2017

We have just learned from Le Bérarde that a new accident has occurred on the Pelvoux plateau: a young man, part of a three-man company from Lyons, has fallen to his death.

– Le Temps, 29 July 1931

Resolution für Prädikatenlogik

Ein konkreter Algorithmus zum logischen Schließen:

(1) Logische Konsequenz auf Unerfüllbarkeit reduzieren

(2) Formeln in Klauselform umwandeln
- Formel bereinigen
- Negationsnormalform bilden
- Pränexform bilden
- Skolemform bilden
- Konjunktive Normalform bilden

(3) Resolutionsverfahren anwenden
- Unifikation zum Finden passender Klauseln
- Bilden von Resolventen bis zur Terminierung

Unifikation

Unifikationsalgorithmus

Eingabe: Unifikationsproblem G
Ausgabe: allgemeinster Unifikator für G, oder „nicht unifizierbar"

Wende die folgenden Umformungsregeln auf G an, bis keine Regel mehr zu einer Änderung führt:

- Löschen: $\{t \doteq t\} \cup G' \rightsquigarrow G'$
- Zerlegung: $\{f(s_1, \ldots, s_n) \doteq f(u_1, \ldots, u_n)\} \cup G' \rightsquigarrow \{s_1 \doteq u_1, \ldots, s_n \doteq u_n\} \cup G'$
- Orientierung: $\{t \doteq x\} \cup G' \rightsquigarrow \{x \doteq t\} \cup G'$ falls $x \in \mathbf{V}$ und $t \notin \mathbf{V}$
- Eliminierung: $\{x \doteq t\} \cup G' \rightsquigarrow \{x \doteq t\} \cup G'\{x \mapsto t\}$ falls $x \in \mathbf{V}$ nicht in t vorkommt

Wenn G dann in gelöster Form ist, dann gib σ_G aus.
Andernfalls gib aus „nicht unifizierbar".

Unifikation von Atomen

Ein Unifikator für eine Menge $\mathcal{A} = \{A_1, \ldots, A_n\}$ von prädikatenlogischen Atomen ist eine Substitution θ mit $A_1\theta = A_2\theta = \ldots = A_n\theta$.

Beobachtungen:

- Eine Menge von Atomen $\mathcal{A}$ ist nur dann unifizierbar, wenn alle Atome das gleiche Prädikat verwenden, d.h. wenn es ein ℓ-stelliges Prädikatensymbol p gibt, so dass $A_i = p(t_{i,1}, \ldots, t_{i,\ell})$ für alle $i \in \{1, \ldots, n\}$.
- Dann ist σ genau dann ein Unifikator für $\mathcal{A}$ wenn σ Unifikator für das folgende Unifikationsproblem $G_{\mathcal{A}}$ ist:

$$\{t_{1,1} \doteq t_{2,1}, \ldots, t_{n-1,1} \doteq t_{n,1}, \ldots, t_{1,\ell} \doteq t_{2,\ell}, \ldots, t_{n-1,\ell} \doteq t_{n,\ell}\}$$

- Insbesondere ist der allgemeinste Unifikator für $G_{\mathcal{A}}$ auch der allgemeinste Unifikator für $\mathcal{A}$

Die Resolutionsregel

Die Resolvente von zwei Klauseln der Form

$$K_1 = \{A_1, \ldots, A_n, L_1, \ldots, L_k\} \text{ und } K_2 = \{\neg A'_1, \ldots, \neg A'_m, L'_1, \ldots, L'_\ell\},$$

für welche σ der allgemeinste Unifikator der Menge $\{A_1, \ldots, A_n, A'_1, \ldots, A'_m\}$ ist und L_i, L'_j beliebige Literale sind, ist die Klausel $\{L_1\sigma, \ldots, L_k\sigma, L'_1\sigma, \ldots, L'_\ell\sigma\}$.

Beispiel: Die Klauseln $K_1 = \{\neg\text{Mensch}(x), \text{hatVater}(x, f(x))\}$ und $K_2 = \{\neg\text{hatVater}(z, v), \text{hatKind}(v, z)\}$ können resolviert werden. Ein allgemeinster Unifikator von $\{\text{hatVater}(x, f(x)), \text{hatVater}(z, v)\}$ ist $\sigma = \{z \mapsto x, v \mapsto f(x)\}$. Die entsprechende Resolvente von K_1 und K_2 ist $\{\neg\text{Mensch}(x), \text{hatKind}(f(x), x)\}$.

Resolution: Beispiel (1)

Wir hatten die folgende Beispielformel F betrachtet:

$$\begin{aligned} &\forall x.((W(x) \wedge \neg L(x)) \vee (L(x) \wedge \neg W(x))) \\ \wedge\ &(\exists x.W(x) \to (\forall x.W(x) \vee \forall x.L(x))) \\ \wedge\ &(\exists x.L(x) \to \neg(\forall x.W(x) \vee \forall x.L(x))) \end{aligned}$$

(Jeder ist Typ W oder Typ L / Ist einer Typ W, dann gibt es hier nur einen Typ / Ist einer Typ L, dann gibt es hier nicht nur einen Typ)

Folgt aus F, dass alle Typ W sind?

Vorgehen:

- Formalisiere diese Frage: $F \models \forall z.W(z)$?
- Reduktion auf Unerfüllbarkeit: Ist $F \wedge \neg\forall z.W(z)$ unerfüllbar?
- Klauselform: F haben wir bereits in Klauselform gebracht. Wir können direkt die Klauseln für $\neg\forall z.W(z)$ hinzufügen:
  - Bereinigte NNF (und Pränexform): $\exists z.\neg W(z)$
  - Skolemform (und KNF): $\neg W(a)$ (a ist Skolemkonstante)

Resolution: Beispiel (2)

Zusammen mit der Klauselform für F erhalten wir die Klauseln:

$$\begin{array}{rlll}
(1) & \{W(x_1), L(x_1)\} & & \\
(2) & \{\neg L(x_1), L(x_1)\} & & \\
(3) & \{W(x_1), \neg W(x_1)\} & & \\
(4) & \{\neg L(x_1), \neg W(x_1)\} & & \\
(5) & \{\neg W(x_2), W(x_3), L(x_4)\} & & \\
(6) & \{\neg L(x_5), \neg W(f_6(x_1, x_2, x_3, x_4, x_5))\} & & \\
(7) & \{\neg L(x_5), \neg L(f_7(x_1, x_2, x_3, x_4, x_5))\} & & \\
(8) & \{\neg W(a)\} & & \\
(9) & \{L(a)\} & (1) + (8) & \{x_1 \mapsto a\} \\
(10) & \{\neg L(f(x_1, x_2, x_3, x_4, a))\} & (9) + (7) & \{x_5 \mapsto a\}
\end{array}$$

Problem:

- $\neg L(f(x_1, x_2, x_3, x_4, a))$ bedeutet „es gibt Nicht-Lügner“ (bezeichnet mit Termen der Form $f(x_1, x_2, x_3, x_4, a)$)
- Dies sollte z.B. mit (1) „Jeder Nicht-Lügner ist Wahrheitssager“ resolvieren
- Aber $\{\neg L(f(x_1, x_2, x_3, x_4, a)), L(x_1)\}$ hat keinen Unifikator

Varianten von Klauseln

Wir wissen: $\forall x.(F \wedge G) \equiv (\forall x.F \wedge \forall x.G)$

In Klauselform kann man sich also die Allquantoren direkt vor jeder einzelnen Klausel denken:

$$\begin{array}{rl} (1) & \forall x_1.\{W(x_1), L(x_1)\} \\ (2) & \forall x_1.\{\neg L(x_1), L(x_1)\} \\ & \ldots \\ (10) & \forall x_1, x_2, x_3, x_4.\{\neg L(f(x_1, x_2, x_3, x_4, a))\} \end{array}$$

Daher darf man die Variablen jeder Klausel einheitlich umbenennen, unabhängig von jeder anderen Klausel, z.B.

$$\{\neg L(f(x_1, x_2, x_3, x_4, a))\} \quad \leadsto \quad \{\neg L(f(x'_1, x'_2, x'_3, x'_4, a))\}$$

Klauseln, die durch eineindeutige Umbenennung von Variablen entstanden sind, nennt man Varianten (einer Klausel)

$\leadsto$ Wir bilden bei der Resolution Varianten um Konflikte von Variablen zu vermeiden

Resolution: Beispiel (3)

Mit einer Variante von Klausel (11) gelingt die Resolution:

$$\begin{array}{rll} (1) & \{W(x_1), L(x_1)\} & \\ (2) & \{\neg L(x_1), L(x_1)\} & \\ (3) & \{W(x_1), \neg W(x_1)\} & \\ (4) & \{\neg L(x_1), \neg W(x_1)\} & \\ (5) & \{\neg W(x_2), W(x_3), L(x_4)\} & \\ (6) & \{\neg L(x_5), \neg W(f_6(x_1, x_2, x_3, x_4, x_5))\} & \\ (7) & \{\neg L(x_5), \neg L(f_7(x_1, x_2, x_3, x_4, x_5))\} & \\ (8) & \{\neg W(a)\} & \\ (9) & \{L(a)\} & (1) + (8)\ \{x_1 \mapsto a\} \\ (10) & \{\neg L(f(x'_1, x'_2, x'_3, x'_4, a))\} & (9) + (7)\ \{x_5 \mapsto a\} \\ (11) & \{W(f(x'_1, x'_2, x'_3, x'_4, a))\} & (1) + (10)\ \{x_1 \mapsto f(x'_1, x'_2, x'_3, x'_4, a)\} \\ (12) & \{W(x_3), L(x_4)\} & (11) + (5)\ \{x_2 \mapsto f(x'_1, x'_2, x'_3, x'_4, a)\} \\ (13) & \{L(x_4)\} & (12) + (8)\ \{x_3 \mapsto a\} \\ (14) & \{\} & (13) + (10)\ \{x_4 \mapsto f(x'_1, x'_2, x'_3, x'_4, a)\} \end{array}$$

Resolution: Beispiel (4)

Wir haben durch Resolution die leere Klausel $\{\}$ abgeleitet

Die leere Klausel bezeichnen wir auch mit $\bot$:

- Sie steht für die leere Disjunktion,
- d.h. für eine falsche (unerfüllbare) Behauptung

$\leadsto$ Wir haben gezeigt, dass die Klauselmenge unerfüllbar ist
$\leadsto$ Die geprüfte logische Konsequenz $F \models \forall z.W(z)$ gilt

Der Resolutionsalgorithmus

Eingabe: Formel F

- Wandle F in Klauselform um $\leadsto$ Klauselmenge $\mathcal{K}_0$
- Für alle $i \geq 0$:
  - $\mathcal{K}_{i+1} := \mathcal{K}_i$
  - Für alle Klauseln $K_1, K_2 \in \mathcal{K}_i$:
    - Bilde von K_1 und K_2 Varianten K'_1 und K'_2, welche keine Variablen gemeinsam haben
    - Bilde alle möglichen Resolventen von K'_1 und K'_2 und füge diese zu $\mathcal{K}_{i+1}$ hinzu
  - Falls $\bot \in \mathcal{K}_{i+1}$, dann terminiere und gib „unerfüllbar“ aus
  - Falls $\mathcal{K}_i = \mathcal{K}_{i+1}$, dann terminiere und gib „erfüllbar“ aus

Anmerkung 1: $K_1 = K_2$ ist erlaubt und manchmal notwendig
Anmerkung 2: $K'_1 = K_1$ und/oder $K'_2 = K_2$ ist möglich, sofern die Varianten keine gemeinsamen Variablen haben

Korrektheit des Resolutionsalgorithmus (1)

Wir wollen den folgenden Satz schrittweise beweisen:

Resolutionssatz: Sei F eine prädikatenlogische Formel und $\mathcal{K}_i$ ($i \geq 0$) die vom Resolutionsalgorithmus ermittelten Klauselmengen. Dann sind die folgenden Aussagen äquivalent:

- F ist unerfüllbar
- Es gibt ein $\ell \geq 0$ mit $\bot \in \mathcal{K}_\ell$

Beweis (Korrektheit): Wir zeigen Korrektheit eines Resolutionsschrittes, dann folgt die Behauptung durch Induktion über die Schrittzahl. Wir unterscheiden Klauseln K vom Satz $\forall K$, für den sie stehen (=Disjunktion mit allquantifizierten Variablen).

Wir hatten bereits erkannt, dass Varianten von Klauseln deren logische Konsequenzen sind (in der Notation des Algorithmus: $\forall K_1 \models \forall K_1'$ und $\forall K_2 \models \forall K_2'$).

Wir zeigen noch die Korrektheit des reinen Resolutionsschrittes.

Korrektheit des Resolutionsalgorithmus (2)

Beweis (Korrektheit, Fortsetzung): Gegeben:

- Klauseln $K_1 = \{A_1, \ldots, A_n, L_1, \ldots, L_k\}$ und $K_2 = \{\neg A_1', \ldots, \neg A_m', L_1', \ldots, L_\ell'\}$
- (allgemeinster) Unifikator σ der Menge $\{A_1, \ldots, A_n, A_1', \ldots, A_m'\}$
- zugehörige Resolvente $K = \{L_1\sigma, \ldots, L_k\sigma, L_1'\sigma, \ldots, L_\ell'\sigma\}$

Sei $\mathcal{I}$ eine beliebige Interpretation.

- Falls $\mathcal{I} \models \forall K_1 \wedge \forall K_2$, dann gilt auch $\mathcal{I} \models \forall(K_1\sigma) \wedge \forall(K_2\sigma)$ (die Substitution konkretisiert eine Allaussage)
- Also gilt für alle Zuweisungen $\mathcal{Z}$: $\mathcal{I}, \mathcal{Z} \models (K_1\sigma) \wedge (K_2\sigma)$
- Fall 1: $\mathcal{I}, \mathcal{Z} \models A_1\sigma$ $(= A_2\sigma = \ldots = A_m'\sigma)$. Dann gilt $\mathcal{I}, \mathcal{Z} \models L_1'\sigma \vee \ldots \vee L_\ell'\sigma$, und damit $\mathcal{I}, \mathcal{Z} \models K$
- Fall 2: $\mathcal{I}, \mathcal{Z} \not\models A_1\sigma$ $(= A_2\sigma = \ldots = A_m'\sigma)$. Dann gilt $\mathcal{I}, \mathcal{Z} \models L_1\sigma \vee \ldots \vee L_k\sigma$, und damit $\mathcal{I}, \mathcal{Z} \models K$
- Also gilt $\mathcal{I} \models \forall K$.

Da $\mathcal{I}$ beliebig ist gilt also $\forall K_1 \wedge \forall K_2 \models \forall K$ – d.h., jede Resolvente ist logische Konsequenz der resolvierten Klauseln

Vollständigkeit des Resolutionsalgorithmus

Resolutionssatz: Sei F eine prädikatenlogische Formel und $\mathcal{K}_i$ ($i \geq 0$) die vom Resolutionsalgorithmus ermittelten Klauselmengen. Dann sind die folgenden Aussagen äquivalent:

- F ist unerfüllbar
- Es gibt ein $\ell \geq 0$ mit $\bot \in \mathcal{K}_\ell$

Bisher gezeigt: Die zweite Aussage impliziert die erste (Korrektheit)

Vollständigkeit ist die Umkehrung

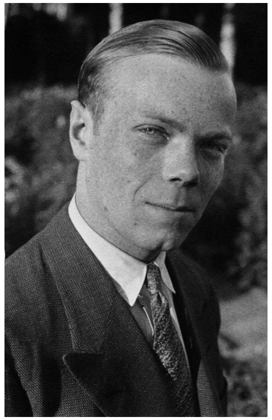
- Jeder Widerspruch wird irgendwann durch Resolution gefunden
- Das ist nicht so offensichtlich – wir müssen dazu etwas weiter ausholen ...

We mentioned yesterday that a young man, part of a company of Alpinists climbing in the area of Le Bérarde, fell to his death. he was M Jacques Herbrand, residing at 10 rue Viollet-le-duc, Paris. M Herbrand departed Sunday with three comrades, MM Jean Brille, Pierre Delair, and Henri Guigner, to ascend the Baus.

During the descent, the piton to which the line was attached gave way, carrying with it a small platform on which M Herbrand was situated, and fell into a chasm.

A rescue party has left to seek the body, which it hopes to reach today.

– Le Temps, 30 July 1931



Jacques Herbrand

12.2.1908 – 27.7.1931

Prädikatenlogische Modelle

Wir wollen zeigen: Wenn es kein Modell für eine Formel gibt, dann leitet Resolution $\perp$ ab

Problem: Modelle sind sehr allgemeine Strukturen

- Beliebige Menge als Domäne
- Systematische Betrachtung schwierig

Idee von Herbrand (und Skolem und Gödel):

„Semantik aus Syntax“

Konstruktion von Modellen direkt aus den Formeln, welche sie erfüllen sollen

Herbranduniversum

Der Kern von Herbrands Idee ist eine „syntaktische“ Domäne:

Sei a eine beliebige Konstante. Das Herbranduniversum Δ_F für eine Formel F ist die Menge aller variablenfreien Terme, die man mit Konstanten und Funktionssymbolen in F und der zusätzlichen Konstante a bilden kann:

- $a \in \Delta_F$
- $c \in \Delta_F$ für jede Konstante aus F
- $f(t_1, \ldots, t_n) \in \Delta_F$ für jedes n-stellige Funktionssymbol aus F und alle Terme $t_1, \ldots, t_n \in \Delta_F$

Anmerkung: Das Herbrand-Universum ist immer abzählbar, manchmal endlich und niemals leer.

Beispiel: Für die Formel $F = p(f(x), y, g(z))$ ergibt sich $\Delta_F = \{a, f(a), g(a), f(f(a)), f(g(a)), g(f(a)), g(g(a)), \ldots\}$.

Herbrandinterpretationen

Mit dem Herbrand-Universum als Domäne kann man Interpretationen definieren, die Terme „ durch sich selbst“ interpretieren:

Eine Herbrandinterpretation für eine Formel F ist eine Interpretation $\mathcal{I}$ für die gilt:

- $\Delta^{\mathcal{I}} = \Delta_F$ ist das Herbrand-Universum von F
- Für jeden Term $t \in \Delta_F$ gilt $t^{\mathcal{I}} = t$

$\mathcal{I}$ ist ein Herbrandmodell für F wenn zudem gilt $\mathcal{I} \models F$.

Anmerkung: Die Definition stellt Bedingungen an Grundbereich und Terminterpretation, aber sie lässt auch viele Freiheiten (z.B. die Interpretation von Prädikatensymbolen)

Beispiel

Betrachten wir wieder die (skolemisierte) Formel
$F = \forall x.\text{hatVater}(x, f(x))$.

Herbranduniversum: $\Delta_F = \{a, f(a), f(f(a)), \ldots\}$

Alle Herbrandinterpretationen stimmen auf der Domäne und (dem relevanten Teil) der Terminterpretation überein.

- $\mathcal{I}_1$ mit $\text{hatVater}^{\mathcal{I}_1} = \emptyset$ ist kein Herbrandmodell
- $\mathcal{I}_2$ mit $\text{hatVater}^{\mathcal{I}_2} = \{\langle t, f(t)\rangle \mid t \in \Delta_F\}$ ist ein Herbrandmodell
- $\mathcal{I}_3$ mit $\text{hatVater}^{\mathcal{I}_3} = \Delta_F \times \Delta_F$ ist ein Herbrandmodell

Syntax vs. Semantik

Bei Herbrandinterpretationen kann man semantische Elemente (wie sie in Zuweisungen vorkommen) durch syntaktishe Elemente (wie sie in Substitutionen vorkommen) ausdrücken:

Lemma: Für jede Herbrandinterpretation $\mathcal{I}$, jede Zuweisung $\mathcal{Z}$ für $\mathcal{I}$, jeden Term $t \in \Delta^{\mathcal{I}}$ und jede Formel F gilt:

$$\mathcal{I}, \mathcal{Z}\{x \mapsto t\} \models F \qquad \text{gdw.} \qquad \mathcal{I}, \mathcal{Z} \models F\{x \mapsto t\}$$

(ohne Beweis, einfach)

Anmerkung: Man kann ein entsprechendes Resultat auch für Nicht-Herbrand-Interpretationen zeigen. Dann muss man einfach den Term auf der linken Seite durch $t^{\mathcal{I},\mathcal{Z}}$ ersetzen.

Erfüllbar + Skolem = Erfüllbarkeit bei Herbrand

Satz: Ein Satz F in Skolemform ist genau dann erfüllbar, wenn F ein Herbrandmodell hat.

Beweis: ($\Leftarrow$) ist klar, da Herbrandmodelle auch Modelle sind.

($\Rightarrow$) Sei $\mathcal{I} \models F$ ein Modell für F. Wir definieren eine Herbrandinterpretation $\mathcal{J}$ indem wir festlegen:

- $p^{\mathcal{J}} = \{\langle t_1, \ldots, t_n\rangle \mid \langle t_1^{\mathcal{I}}, \ldots, t_n^{\mathcal{I}}\rangle \in p^{\mathcal{I}}\}$
  Anm.: t_i sind variablenfrei, daher ist $t_i^{\mathcal{I}}$ wohldefiniert

Behauptung: $\mathcal{J}$ ist ein Herbrandmodell von F

Beweis (Fortsetzung)

Behauptung: $\mathcal{J}$ ist ein Herbrandmodell von F

F hat die Form $\forall x_1, \ldots, x_n.G$, wobei G quantorenfrei ist.

- Aus $\mathcal{I} \models F$ folgt also $\mathcal{I}, \mathcal{Z} \models G$ für jede Zuweisung $\mathcal{Z}$ für $\mathcal{I}$
- Speziell gilt also für alle $t_1, \ldots, t_n \in \Delta_F$:
  $\mathcal{I}, \{x_1 \mapsto t_1^{\mathcal{I}}, \ldots, x_n \mapsto t_n^{\mathcal{I}}\} \models G$
- Daraus folgt: $\mathcal{I} \models G\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ (analog zu Lemma)
- Daraus folgt: $\mathcal{J} \models G\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$
  (Direkt aus Definition für Atome G; die Aussage kann leicht auf größere Boolsche Verknüpfungen von Atomen verallgemeinert werden – formal durch strukturelle Induktion)
- Es folgt: $\mathcal{J}, \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} \models G$ (Lemma)

Der Schluss gilt für alle $t_1, \ldots, t_n \in \Delta_F$, d.h. $\mathcal{J} \models F$. $\square$

Gegenbeispiel

Der Satz gilt nicht unbedingt, wenn Formeln nicht in Skolemform sind:

Beispiel: Die folgende Formel ist offensichtlich erfüllbar:

$$\exists x.p(x) \wedge \exists y.\neg p(y)$$

Die Formel verwendet aber keine Funktionen oder Konstanten
$\rightsquigarrow$ das Herbrand-Universum ist $\{a\}$

Aber keine Interpretation $\mathcal{I}$ mit Domäne $\{a\}$ ist Modell der Formel, da in diesem Fall entweder $p^{\mathcal{I}} = \emptyset$ oder $(\neg p)^{\mathcal{I}} = \emptyset$ ist.

Zum Vergleich die Skolemform der Formel dieses Beispiels:

$$p(c) \wedge \neg p(d)$$

Hier gibt es zwei (Skolem-)Konstanten im Herbrand-Universum
$\rightsquigarrow$ Es gibt ein Herbrand-Modell mit dieser Domäne

Löwenheim-Skolem

Satz von Löwenheim[1] und Skolem: Jede erfüllbare prädiktenlogische Formel hat ein abzählbares Modell (d.h. eines mit abzählbarer Domäne).

Beweis:

- Jede Formel F kann in erfüllbarkeitsäquivalente Skolemform F' überführt werden
- Ist F erfüllbar, so hat F' ein (abzählbares) Herbrandmodell
- Man kann aus den Beweisen der Erfüllbarkeitsäquivalenz erkennen: jedes Modell von F' ist auch ein Modell für F □

[1] Leopold Löwenheim[2] (1878–1957): deutscher Logiker; 1915 erster Beweis des obigen Satzes; 1916 Soldat im 1. Weltkrieg; 1934 von den Nazis zwangspensioniert und als Privatlehrer tätig
[2] „Note: One of the best predictors of success in mathematical logic is having an umlaut in your name" – S. Aaronson, Quantum Computing since Democritus. Cambridge, 2013.

Historische Anmerkungen

- Der eben gezeigte „Löwenheim-Skolem-Satz" ist eigentlich der nach unten gerichtete Satz von Löwenheim und Skolem (Downward Löwenheim-Skolem Theorem)
- Der nach oben gerichtete Satz von Löwenheim und Skolem (Upward Löwenheim-Skolem Theorem) besagt, dass jede erfüllbare Formel auch in Modellen beliebig großer Kardinalität erfüllbar ist
- Der Legende nach war Skolem die Verwendung seines Namens in diesem Kontext unangenehm:

> „I follow custom in calling Corollary 6.1.4 the upward Löwenheim-Skolem theorem. But in fact Skolem didn't even believe it, because he didn't believe in the existence of uncountable sets."
> – W. Hodges, Model theory, Cambridge 1993

Zusammenfassung und Ausblick

Die prädikatenlogische Resolution ist ein Semi-Entscheidungsverfahren für die Unerfüllbarkeit logischer Formeln

Man kann Erfüllbarkeit auf Erfüllbarkeit über „syntaktisch definierten" Herbrandmodellen reduzieren

Prädikatenlogik kann verschiedene Unendlichkeiten nicht unterscheiden: Eine Theorie hat beliebig große unendliche Modelle genau dann wenn sie abzählbare Modelle hat

Was erwartet uns als nächstes?

- Beweis der Vollständigkeit der Resolution
- Logik über endlichen Modellen und ihre praktische Anwendung
- Gödel

Bildrechte

Folie 17: Fotografie von Natasha Artin Brunswick, 1931, CC-By 3.0