Kombinatorische Logik interaktiv

Skifahren mit der Logic Workbanch

Burkhardt Renz

29. Juni 2021

Fachbereich MNI
Technische Hochschule Mittelhessen

Moses Schönfinkel 1888–1942

Über die Bausteine der mathematischen Logik.

Von

M. Schönfinkel in Moskau[1]).

§ 1.

Es entspricht dem Wesen der axiomatischen Methode, wie sie heute vor allem durch die Arbeiten Hilberts zur Anerkennung gelangt ist, daß man nicht allein hinsichtlich der Zahl und des Gehalts der *Axiome* nach möglichster Beschränkung strebt, sondern auch die Anzahl der als undefiniert zugrunde zu legenden *Begriffe* so klein wie möglich zu machen sucht, indem man nach Begriffen fahndet, die vorzugsweise geeignet sind, um aus ihnen alle anderen Begriffe des fraglichen Wissenszweiges aufzubauen. Begreiflicherweise wird man sich im Sinne dieser Aufgabe bezüglich des Verlangens nach Einfachheit der an den Anfang zu stellenden Begriffe entsprechend bescheiden müssen.

Bekanntlich lassen sich die grundlegenden *Aussagenverknüpfungen* der mathematischen Logik, die ich hier in der von Hilbert in seinen Vorlesungen verwendeten Bezeichnungsweise wiedergebe:

$$\bar{a}, \quad a \vee b, \quad a \,\&\, b, \quad a \rightarrow b, \quad a \sim b$$

(lies: „a nicht“, „a oder b“, „a und b“, „wenn a, so b“, „a äquivalent b“), aus einer einzigen von ihnen überhaupt nicht, aus zweien von ihnen aber nur in der Weise gewinnen, daß man die Negation und irgendeine der drei folgenden Verknüpfungen als undefiniert zugrunde legt. (Von diesen drei Arten der Zurückführung haben Whitehead und Russell die erste und Frege die dritte verwendet.)

Daß dessenungeachtet die Zurückführung auf eine einzige Grundverknüpfung sehr wohl möglich ist, sobald man sich von der Einschränkung

[1]) Die folgenden Gedanken wurden vom Verfasser am 7. Dez. 1920 vor der Mathematischen Gesellschaft in Göttingen vorgetragen. Ihre formale und stilistische Durcharbeitung für diese Veröffentlichung wurde von H. Behmann in Göttingen übernommen.

Prädikatenlogik ohne Variablen

- Henry M. Sheffer 1913: *Die Aussagenlogik kann mit einem einzigen Junktor definiert werden* — dem `nand`, $\uparrow$.
- Kann man das auch für die Prädikatenlogik?
- Schönfinkels Resultat: ja, mit **C**, **S** und **U**[1]
- Beispiel:
  $\forall n \exists p (N(n) \to (P(p) \wedge G(p,n)) \mathrel{\hat{=}}$
  $\mathbf{U}N(\mathbf{B}(\mathbf{U}P)(\mathbf{C}G))\mathbf{U}N(\mathbf{B}(\mathbf{U}P)(\mathbf{C}G))$,
  wobei sich **B** aus **C** und **S** „kombinieren“ lässt.

[1] „Es besteht nun die bemerkenswerte Tatsache, daß jede logische Formel sich allein ... durch C, S und U ausdrücken läßt.“ Bausteine der mathematischen Logik, 1924

Schönfinkels Funktionenkalkül

- Mit Funktionen höherer Ordnung kann man Funktionen mit mehreren Variablen auf unäre Funktionen reduzieren: $F(x, y) = G_x(y)$ wobei die Funktion G_x selbst durch eine Funktion $G = f(x)$ gebildet wird.
- „Es sollen nun eine Reihe von *individuellen Funktionen* von sehr allgemeiner Natur eingeführt werden“:
Identitätsfunktion	$\mathbf{I}x \triangleright x$
Konstanzfunktion[2]	$\mathbf{K}xy \triangleright x$
Verschmelzungsfunktion	$\mathbf{S}\phi\chi x \triangleright (\phi x)(\chi x)$

[2] hieß bei Schönfunkel **C**

Kombinatorische Logik als formale Sprache

- Die Objekte der Sprache sind *Terme*.
- Primitive Terme sind *Platzhalter* und Konstanten.
- Terme werden induktiv definiert:
  1. Platzhalter $x, y, \ldots$ sind Terme
  2. Konstanten $\mathbf{S}, \mathbf{K}, \ldots$ sind Terme
  3. Sind M und N Terme, dann ist auch (MN) ein Term
- Um Klammern zu sparen, wird die *Applikation* (MN) links-assoziativ interpretiert, d.h. $MNP \mathrel{\hat{=}} ((MN)P)$.
- Beispiele: $\mathbf{S}\ \mathbf{K}\ \mathbf{K}$

  $\mathbf{S}\ \mathbf{I}\ \mathbf{I}(\mathbf{S}\ \mathbf{I}\ \mathbf{I})$

  $\mathbf{S}(\mathbf{S}xyz)(\mathbf{S}xyz)z$

Kombinatorische Logik als Berechnungsmodell

- Wir fassen Applikation als *Funktionsapplikation* auf.
- Wir fassen Kombinatoren (= Terme ohne Platzhalter) als Funktionen auf.
Identitätsfunktion	$\mathbf{I}x \triangleright x$
Konstanzfunktion	$\mathbf{K}xy \triangleright x$
Verschmelzungsfunktion	$\mathbf{S}xyz \triangleright xz(yz)$
- *Reduktionsschritt*: Ersetze *Redex* durch seinen Effekt
- *Expansionsschritt*: Ersetze einen Subterm durch einen Redex, der ihn erzeugt.

Kombinatorische Logik in der Logic Workbench, I

- Syntax
- Definition der Kombinatoren
- Interaktive Session
- Reduktion
- Expansion

Haskell Brooks Curry 1900–1982

Grundlagen der kombinatorischen Logik.

TEIL I.

von H. B. CURRY.

INHALTSÜBERSICHT.

KAPITEL I. ALLGEMEINE GRUNDLAGEN.

A. EINLEITUNG.

Im Anfang jeder mathematisch-logischen Untersuchung setzt man eine gewisse Menge von Kategorien voraus, die als ein Teil des irreduziblen Minimums von Kenntnis, womit man anzufangen hat, betrachtet werden sollen, Als solche Kategorien gelten gewöhnlich Aussagen und Aussagefunktionen von verschiedenen Ordnungen und Stufen. Diese Kategorien müssen weiterhin nicht nur als rein formale Begriffe vorausgesetzt werden, sondern es muss zuweilen einen Sinn haben, dass ein gegebener Gegenstand zu dieser oder jener Kategorie gehört. Mit anderen Worten, sie müssen eine inhaltliche Eigenschaft besitzen, nämlich die, dass sie Kategorien sind; und so sind sie das, was ich schon nicht-formale Grundbegriffe genannt habe.†

* Die Abschnitte *C, D, E* werden in Teil II erscheinen.

† In einer Abhandlung, "An Analysis of Logical Substitution," *American Journal of Mathematics*, Bd. 51 (Juli, 1929), S. 363-384.

509

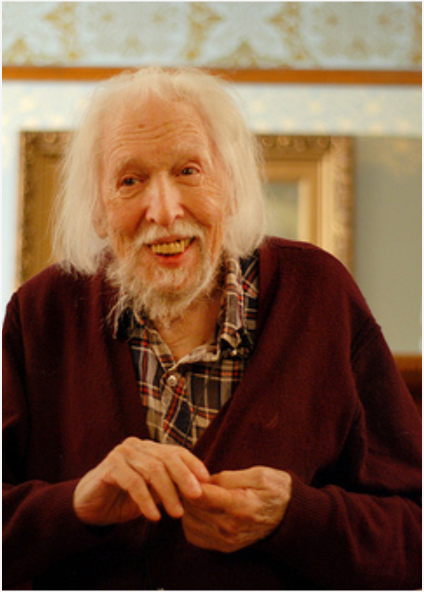
Eigenschaften der kombinatorischen Logik

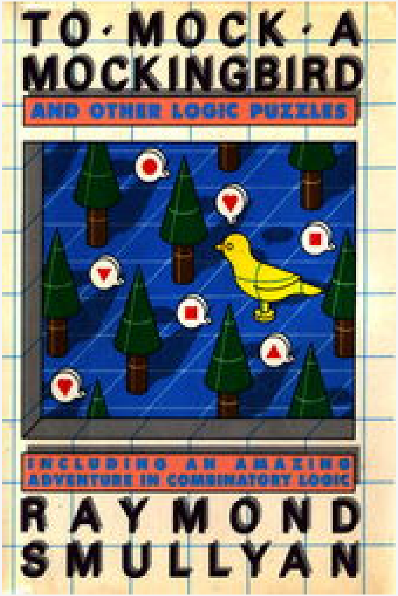
- Ein Term ist in *Normalform*, wenn er nicht mehr reduziert werden kann.
- Church-Rosser: Wenn ein Term eine Normalform hat, dann ist sie eindeutig.
- Terme können denselben Effekt haben, auch wenn sie nicht zueinander reduzierbar sind.
- Fixpunkt-Satz: Für alle Terme M gibt einen Term N mit $MN = N$.
- Kombinatorische Vollständigkeit: Jede explizit definierbare Funktion kann als Kombinator repräsentiert werden – *bracket abstraction*
- Die kombinatorische Logik ist *unentscheidbar*.

Kombinatorische Logik in der Logic Workbench, II

- Interessante Beispiele
- Normalform
- Church-Rosser
- Currys Fixpunkt-Kombinator
- *Bracket Abstraction*

Raymond Smullyan 1919–2017





Vogelgezwitscher

„A certain enchanted forest is inhabited by talking birds. Given any birds A and B, if you call out the name of B to A, then A will respond by calling out the name of some bird to you; this bird we designate by AB."

„Given any birds, A, B, and C (not necessarily distinct) the bird C is said to *compose* A with B if for every bird x the following condition holds:

$$Cx = A(Bx)$$

In words, this means, that C's response to x is the same as A's response to B's response to x."

Kombinatorische Logik in der Logic Workbench, III

- Smullyans Rätsel interaktiv lösen

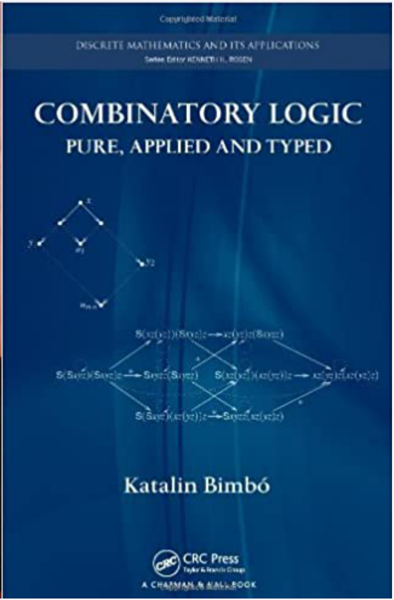
Smullyans „intelligentere“ Vögel

- Logical Birds (Chap. 23)
  $\mathbf{K} \mathrel{\hat{=}} \texttt{true}$, $\mathbf{K}\ \mathbf{I} \mathrel{\hat{=}} \texttt{false}$, $\mathbf{S}\ (\mathbf{S}\ \mathbf{I}\ (\mathbf{K}\ (\mathbf{K}\ \mathbf{I})))\ (\ \mathbf{K}\ \mathbf{K}) \mathrel{\hat{=}} \neg$
- Birds That Can Do Arithmetic (Chap. 24)
  $\mathbf{K}\ \mathbf{I} \mathrel{\hat{=}} 0$, $\mathbf{S}\ \mathbf{B} \mathrel{\hat{=}}$ Nachfolgerfunktion
  $\mathbf{S}\ \mathbf{I}\ (\ \mathbf{K}\ (\mathbf{S}\ \mathbf{B})) \mathrel{\hat{=}}$ Addition

Kombinatorische Logik in der Logic Workbench, IV

- Boolesche Werte und Operatoren der Aussagenlogik
- Natürliche Zahlen: Church-Numerale

Katalin Bimbó (University of Alberta)



Das Titelbild des Buches

$$
\begin{tikzcd}
 & \mathbf{S}(xz(yz)(\mathbf{S}xyz)z \arrow[r, "1"] \arrow[dr, "2"] & xz(yz)z(\mathbf{S}xyzz) \arrow[dr, "1"] & \\
\mathbf{S}(\mathbf{S}xyz)(\mathbf{S}xyz)z \arrow[ur, "2"] \arrow[r, "1"] \arrow[dr, "3"] & \mathbf{S}xyzz(\mathbf{S}xyzz) \arrow[ur, "1"] \arrow[dr, "2"] & \mathbf{S}(xz(yz))(xz(yz))z \arrow[r, "1"] & xz(yz)z)xz(yz)z) \\
 & \mathbf{S}(\mathbf{S}xyz)(xy)(yz))z \arrow[ur, "2"] \arrow[r, "1"] & \mathbf{S}xyzz(xz(yz)z) \arrow[ur, "1"] &
\end{tikzcd}
$$

Kombinatorische Logik in der Logic Workbench, V

- Das Titelbild in der Logic Workbench