Logik und Computerbeweise
Schnupperuni 2007

Holger Bluhm,
Prof. Dr. Martin Kreuzer,
Stefan Kühling

02.08.2007

Inhaltsverzeichnis

Logik und Computerbeweise

Holger Bluhm,

Prof. Dr. Martin Kreuzer,

Stefan Kühling

Ja, was ist Logik nun wirklich?

So genau können wir dies leider auch nicht beschreiben. Das Wort „Logik“ entstammt dem altgriechischen Wort „logos“, welches „Vernunft“ bedeutet. Im allgemeinen Sprachgebrauch bedeutet Logik die Fähigkeit, folgerichtig (also „logisch“) zu denken. In den nächsten anderthalb Stunden widmen wir uns hauptsächlich der *formalen Logik*, also dem Studium der formalen Beziehungen zwischen Denkinhalten. Der Prototyp einer mathematischen Beschreibung solcher Denkinhalte ist der der *Aussage*, für die wir die folgende Pseudodefinition anzubieten haben.

Definition 1.1

Eine (logische) **Aussage** ist ein sprachliches Gebilde, das entweder **wahr** oder **falsch** ist.

Beispiel 1.2

Die folgenden sprachlichen Gebilde stellen Beispiele für Aussagen dar.

A = „Berlin ist die Hauptstadt von Deutschland."
B = „Der Wal ist ein Fisch."
C = „Für jede Zahl $x \in \mathbb{R}$ gilt $x^2 \geq 0$."
D = „Das Programm `MyProg(...)` terminiert."

Hingegen stellen die folgenden sprachlichen Gebilde „Aussagen über Aussagen" dar.

E = „Die Aussagen A und B sind beide wahr."
F = „Die Aussage B gilt nicht."
= „Der Wal ist kein Fisch."

Das sprachliche Gebilde „Morgen wird es regnen“ ist jedoch keine Aussage im Sinn von Definition 1.1, denn ihr Wahrheitswert hängt vom Standpunkt des Betrachters ab.

Nachdem wir jetzt die Abkürzungen (die sog. *atomaren Formeln*) haben, bleibt die Frage, was machen wir mit ihnen.

Um kompliziertere, zusammengesetzte Aussagen in logische Formeln zu übersetzen, brauchen wir *Verbindungselemente*.

- „A und B“: $A \wedge B$
- „A oder B“: $A \vee B$
- „nicht A“: $\neg A$

und daraus abgeleitet

- Folgerung „B gilt, wenn A gilt“ oder „A gilt nur dann, wenn B gilt“:

$$A \Rightarrow B \equiv \neg A \vee B$$

- Äquivalenz „B gilt genau dann, wenn A gilt“:

$$A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (A \Leftarrow B) \equiv (\neg A \vee B) \wedge (A \vee \neg B)$$

Wie erhalten wir jetzt den Wahrheitswert einer größeren Formel?

Da wir nur die Wahrheitswerte der elementaren Formeln haben (eine Aussage gilt oder sie gilt nicht), müssen wir den Wahrheitswert einer „größeren“ Formel auf diese zurückführen.

Starten wir also mit der „und“–Verknüpfung:

$\alpha(A)$	$\alpha(B)$	$\alpha(A \wedge B)$
w	w	w
w	f	f
f	w	f
f	f	f

oder

$\alpha(A)$	$\alpha(B)$	$\alpha(A \wedge B)$
1	1	1
1	0	0
0	1	0
0	0	0

Logik und Computerbeweise

Holger Bluhm,

Prof. Dr. Martin Kreuzer,

Stefan Kühling

Für die „oder“–Verknüpfung ergibt sich folgende Wahrheitstabelle.

$\alpha(A)$	$\alpha(B)$	$\alpha(A \vee B)$
1	1	1
1	0	1
0	1	1
0	0	0

Und die „nicht“–Operation ergibt

$\alpha(A)$	$\alpha(\neg A)$
1	0
0	1

Somit ergibt sich für die Formel

$$F := (P \Leftrightarrow \neg S) \wedge (\neg S \vee A) \wedge (\neg A \vee P)$$

folgende Wahrheitstabelle.

$\alpha(P)$	$\alpha(S)$	$\alpha(A)$	$\alpha(P \Leftrightarrow \neg S)$	$\alpha(\neg S \vee A)$	$\alpha(\neg A \vee P)$	$\alpha(F)$
1	1	1	0	1	1	0
1	1	0	0	0	1	0
1	**0**	**1**	1	1	1	**1**
1	**0**	**0**	1	1	1	**1**
0	1	1	1	1	0	0
0	1	0	1	0	1	0
0	0	1	0	1	0	0
0	0	0	0	1	1	0

Also sind die Belegungen $\alpha(P) = \alpha(A) = 1$, $\alpha(S) = 0$ und $\alpha(P) = 1$, $\alpha(S) = \alpha(A) = 0$ Modelle für F.

Wenn man nachweisen möchte, dass eine Formel immer gilt oder besser, dass eine Behauptung immer aus einer Menge von Voraussetzungen folgt, zeigt man die Unerfüllbarkeit der Negation der Formel.

$$F \text{ gilt immer} \quad \Leftrightarrow \quad \neg F \text{ ist unerfüllbar}$$

$$\begin{array}{rcl} & & \neg\,((F_1 \wedge \ldots \wedge F_n) \Rightarrow G) \\ \text{Aus } F_1 \wedge F_2 \wedge \ldots \wedge F_n & \Leftrightarrow & \quad \equiv \neg\,(\neg\,(F_1 \wedge \ldots \wedge F_n) \vee G) \\ \text{folgt (immer) } G & & \quad \equiv (F_1 \wedge \ldots \wedge F_n) \wedge \neg G \\ & & \text{ist unerfüllbar} \end{array}$$

Beispiel 1.3 (Philosophisches)

Platon hatte Recht mit seiner Einschätzung des Sokrates genau dann, wenn Sokrates kein großer Philosoph war. Wenn Sokrates ein großer Philosoph war, dann hatte Aristoteles Recht mit seiner Einschätzung des Platon. Aristoteles hatte nur dann Recht mit seiner Einschätzung des Platon, falls Platon Recht hatte mit seiner Einschätzung des Sokrates.

1) Festlegung der Abkürzungen:

S = „Sokrates war ein großer Philosoph."

A = „Aristoteles hat Recht mit seiner Einschätzung des Platon."

P = „Platon hatte Recht mit seiner Einschätzung des Sokrates."

2) Übersetzung der Aussagen:

- „Platon hatte Recht mit seiner Einschätzung des Sokrates genau dann, wenn Sokrates kein großer Philosoph war.“

$$F_1 := (P \Leftrightarrow \neg S)$$

- „Wenn Sokrates ein großer Philosoph war, dann hatte Aristoteles Recht mit seiner Einschätzung des Platon.“

$$F_2 := (S \Rightarrow A) \equiv (\neg S \vee A)$$

- „Aristoteles hatte nur dann Recht mit seiner Einschätzung des Platon, falls Platon Recht hatte mit seiner Einschätzung des Sokrates.“

$$F_3 := (A \Rightarrow P) \equiv (\neg A \vee P)$$

- Alle Aussagen zusammen werden beschrieben durch die Formel

$$F := F_1 \wedge F_2 \wedge F_3.$$

Computeralgebra

Um jetzt unsere Formeln in eine dem Computer verträgliche Sprache zu übersetzen, brauchen wir einen speziellen Körper (Rechenraum).

Wir verwenden $\mathbb{F}_2$, dieser Körper wird durch die Nullstellengleichung $\left(x^2 - x\right) = 0$ bestimmt. Diese Gleichung hat nur 0 und 1 als mögliche Lösungen. Deshalb sind auch 0 und 1 in diesem Rechenraum die einzigen beiden Elemente. Anders ausgedrückt, wir teilen die ganzen Zahlen in die geraden und ungeraden Zahlen ein und schreiben 0 für die geraden Zahlen und 1 für die ungeraden Zahlen.

Damit verbunden ergeben sich einige spezielle Rechenregeln:

$$1 \cdot 1 = 1 \qquad 1 \cdot 0 = 0 \qquad 1 + 0 = 1 \qquad 1 + 1 = 0$$

oder allgemeiner

$$-x = x \qquad x^2 = x.$$

Wir übersetzen die „Suche nach der Erfüllbarkeit“ in eine „Suche von Nullstellen“.

Logik	Nullstellensuche
Aussagensymbole	Variablen
$A, B, C \quad \text{oder} \quad A_1, \ldots, A_n$	$x, y, z \quad \text{oder} \quad x_1, \ldots, x_n$
Erfüllbarkeit von	Nullstelle von
$A \quad \text{bzw.} \quad B$	$(x-1) =: F \quad \text{bzw.} \quad (y-1) =: G$
$\neg A$	$F + 1 = x$
$A \vee B$	$F \cdot G = (x-1)\,(y-1)$
$A \wedge B \equiv \neg(\neg A \vee \neg B)$	$(F+1)\,(G+1) + 1 = (xy+1)$

Beispiel 2.1

Nehmen wir z.B. die unerfüllbare Formel $A \wedge \neg A$, die Aussagen A und $\neg A$ haben dann die Übersetzungen

$$(x-1) \quad \text{und} \quad ((x-1)+1) = x.$$

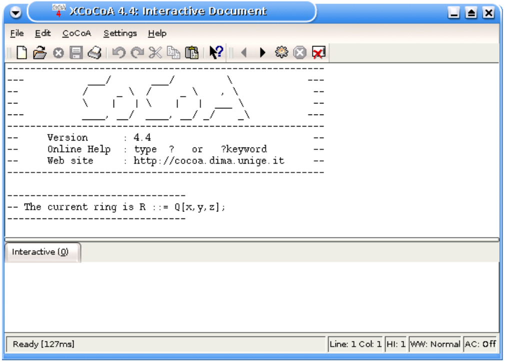
Damit hat dann die Formel $A \wedge \neg A$ die Übersetzung

$$\begin{aligned} ((x-1)+1)\cdot(x+1)+1 &= x\cdot(x+1)+1 \\ &= x^2+x+1 \\ &= x^2-x+1. \end{aligned}$$

Dieses Polynom hat keine Nullstelle, da es sich um die Körpergleichung $\left(x^2-x\right)=0$ um eins verschoben handelt.

Natürlich ist diese Übersetzung mit anschließender Vereinfachung und Nullstellentest per Hand langsamer, als die Wahrheitstafelmethode.

Jedoch haben wir für die Vereinfachungen und Nullstellensuche das Computeralgebra–System CoCoA zur Verfügung.



Das Programm wird mit „xcocoa“ gestartet.

CoCoA–Befehle werden immer groß geschrieben und mit einem Semikolon beendet.

So legt dann `Use Z/(2)[x,y,z];` fest, dass wir im $\mathbb{F}_2$ rechnen und die Variablen x, y, z zur Verfügung haben. Ebenso verursacht z.B. `Use Z/(2)[x[1..6]];` die Festlegung $x_1, \ldots, x_6$.

Als erstes setzen wir immer

```
K:=Ideal(x^2-x,y^2-y,z^2-z);
```

oder

```
K:=Ideal(x[1]^2-x[1],...,x[6]^2-x[6]);
```

Dieses *Körperideal* brauchen wir später.

Logik und Computerbeweise

Holger Bluhm,

Prof. Dr. Martin Kreuzer,

Stefan Kühling

Für das Aussagensymbol A verwenden wir das Polynom $x - 1$ und dieses geben wir als `I1:=Ideal(x-1);` in CoCoA ein.

Natürlich können so auch bereits übersetzte, größere Polynome eingegeben/verwendet werden.

Mit A,B, eingegeben als `I1`,`I2`, können wir auch

$A \wedge B$	mit	`J1:=I1+I2;`
$A \vee B$	mit	`J2:=Intersection(I1,I2);`
$\neg A$	mit	`J3:=K:I1;`

eingeben.

Mit `ReduceGBasis(J);` wird dann das Ideal `J` ausgewertet.

Logik und Computerbeweise
Holger Bluhm,
Prof. Dr. Martin Kreuzer,
Stefan Kühling
Aussagenlogik
Elementares
Wahrheitswerte
Übersetzungen
Computeralgebra
Übersetzungen
CoCoA
Einführung
Logik–Befehle
Computerbeweise

Als Ausgabe erhalten wir dann (hier nur für eine Variable x)

$$
\begin{array}{rcl}
[1] & \widehat{=} & \text{die Formel ist unerfüllbar} \\
[x] & \widehat{=} & \text{die Formel gilt für } x = 0\text{, also für } \alpha(A) = 0 \\
[x+1] & \widehat{=} & \text{die Formel gilt für } x = 1\text{, also für } \alpha(A) = 1 \\
[x\hat{\ }2+x]^{*} & \widehat{=} & \text{die Formel gilt immer (Tautologie)}
\end{array}
$$

* $x^2 + x = (x+1) \cdot x$

Beispiel 4.1

Wir wollen die Formel

$$\begin{aligned} F &:= (P \Leftrightarrow \neg S) \wedge (\neg S \vee A) \wedge (\neg A \vee P) \\ &\equiv (\neg P \vee \neg S) \wedge (P \vee S) \wedge (\neg S \vee A) \wedge (\neg A \vee P) \end{aligned}$$

auf Erfüllbarkeit testen.

Wir verwenden die Variablen x für P, y für S und z für A:

```
Use Z/(2)[x,y,z];
K:=Ideal(x^2-x,y^2-y,z^2-z);
```

Um das Eingegebene besser mit der Formel vergleichen zu können, verwenden wir

`P:=Ideal(x-1);` für P,

`S:=Ideal(y-1);` für S,

`A:=Ideal(z-1);` für A.

$$F \quad = \quad (\neg P \vee \neg S) \wedge (P \vee S) \wedge (\neg S \vee A) \wedge (\neg A \vee P)$$

```
F  :=  Intersection(K:P,K:S)
          + Intersection(P,S)
          + Intersection(K:S,A)
          + Intersection(K:A,P);
```

Wertet man F mit `ReducedGBasis(F);` aus, so erhält man

```
[x + 1, y, z^2 + z],
```

was unserem bekannten Ergebnis von Folie 7 entspricht:

$$\alpha(P) = 1,\ \alpha(S) = 0 \text{ und } \alpha(A) \text{ ist beliebig.}$$