Unvollständigkeit der Arithmetik

Hans U. Simon (RUB)

Email: simon@lmi.rub.de

Homepage: http://www.ruhr-uni-bochum.de/lmi

Zeichenvorrat für arithmetische Formeln

Konstanten:	0	1	2	$\cdots$
Variablen:	x_0	x_1	x_2	$\cdots$
Klammern:	$($	$)$		
arithmetische Verknüpfungen:	$+$	$*$		
Gleichheitszeichen:	$=$			
aussagenlogische Verknüpfungen:	$\neg$	$\vee$	$\wedge$	
Quantoren:	$\exists$	$\forall$		

Syntax arithmetischer Formeln

Terme:

1. Die Konstanten $0, 1, 2, \ldots$ und die Variablen $x_0, x_1, x_2, \ldots$ sind (atomare) Terme.

2. Für Terme t_1, t_2 sind auch $(t_1 + t_2)$ und $(t_1 * t_2)$ Terme.

Formeln:

1. Jede Gleichung $(t_1 = t_2)$ für Terme t_1, t_2 ist eine (atomare) Formel.

2. Für Formeln F, G sind auch $\neg F$, $(F \vee G)$ und $(F \wedge G)$ Formeln.

3. Für eine Formel F und eine Variable x sind auch $\exists x F$ und $\forall x F$ Formeln. F heißt dann der Wirkungsbereich des Quantors “$\exists$” bzw. “$\forall$”.

Beispiele für Formeln

$$\forall x \exists y ((x + y) = (x * (x + 1)))$$

$$\forall x ((x = 0) \vee \exists y ((x * y) = 1))$$

Vereinbarung Wenn die Interpretation dadurch nicht beinträchtigt wird, verzichten wir auf vollständige Klammerung von Formeln.

Mit unvollständiger Klammerung lesen sich die obigen Formeln wie folgt:

$$\forall x \exists y (x + y = x * (x + 1))$$

$$\forall x (x = 0 \vee \exists y (x * y = 1))$$

Auswertung von Termen

Eine Variablenbelegung

$$\phi : V \to \mathbb{N} \text{ mit } V = \{x_0, x_1, x_2, \ldots\}$$

kann zu einer Belegung aller Terme fortgesetzt werden wie folgt:

1. $\phi(n) := n$ für jede Konstante $n \in \mathbb{N}$.
2. $\phi(t_1 + t_2) := \phi(t_1) + \phi(t_2)$ und $\phi(t_1 * t_2) := \phi(t_1)\phi(t_2)$.

Jedem Term wird auf diese Weise ein Wert zugeordnet.

Wenn wir x mit 10 und y mit 8 belegen, ergibt sich zum Beispiel

$$\phi(x + 5 * y) = 10 + 5 \cdot 8 = 50 \ .$$

Freie und gebundene Variablen

Ein Vorkommen von x in F heißt gebunden, falls es im Wirkungsbereich eines Quantors liegt; andernfalls heißt ein Vorkommen von x in F frei.

Eine Variable kann in einer Formel sowohl frei wie gebunden vorkommen. Die Menge der freien Variablen in F ist die Menge der Variablen, die in F mindestens einmal frei vorkommen.

Schreibweise $F(x_1, \ldots, x_k)$ drückt aus, dass $x_1, \ldots, x_k$ die freien Variablen in F sind. Für Konstanten $n_1, \ldots, n_k$ bezeichnet dann

$$F(x_1/n_1, \ldots, x_k/n_k) \text{ bzw. einfach } F(n_1, \ldots, n_k)$$

die Formel, welche aus $F(x_1, \ldots, x_k)$ entsteht, wenn jedes freie Vorkommen von x_i durch n_i ersetzt wird.

Wahre Formeln

Induktive Definition:

1. $(t_1 = t_2)$ ist wahr, falls $\phi(t_1) = \phi(t_2)$ für alle Belegungen $\phi : V \to \mathbb{N}$.
2. $\neg F$ ist wahr, falls F nicht wahr ist.
3. $(F \vee G)$ ist wahr, falls F oder G wahr ist.
4. $(F \wedge G)$ ist wahr, falls F und G wahr sind.
5. $\exists x F$ ist wahr, falls eine Konstante $n \in \mathbb{N}$ existiert, so das $F(x/n)$ wahr ist.
6. $\forall x F$ ist wahr, falls $F(x/n)$ für alle Konstanten $n \in \mathbb{N}$ wahr ist.

Statt „nicht wahr“ sagen wir auch „falsch“.

Beispiele

Die Formel

$$\forall x \exists y (x + y = x * (x + 1))$$

ist wahr! Wähle nämlich $y = x \cdot x$.

Die Formel

$$\forall x (x = 0 \vee \exists y (x * y = 1))$$

ist falsch, da zum Beispiel $x = 2$ in $\mathbb{N}$ kein multiplikatives Inverses besitzt. (Über $\mathbb{Q}$ wäre die Formel wahr.)

„Syntaktischer Zucker“

Erweiterte Syntax	**Reduktion auf „alte“ Syntax**
$F \rightarrow G$ (Implikation)	$\neg F \vee G$
$F \leftrightarrow G$ (Äquivalenz)	$(F \rightarrow G) \wedge (G \rightarrow F)$
$a \leq b$ (Kleinergleich-Relation)	$\exists c(a + c = b)$
$a < b$ (Kleiner-Relation)	$\exists c(a + 1 + c = b)$
$\exists x < a\ F$ (beschränkter Existenzquantor)	$\exists x(x < a \wedge F)$
$\forall x < a\ F$ (beschränkter Allquantor)	$\forall x(x < a \rightarrow F)$

- Analog lassen sich die Relationen „$>, \geq$“ einführen.
- Beschränkte Quantifizierung ist natürlich auch über die Relationen „$>, \leq, \geq$“ möglich.

Arithmetisch repräsentierbare Funktionen

Eine Funktion $f : \mathbb{N}^k \rightarrow \mathbb{N}$ heißt arithmetisch repräsentierbar, falls es eine arithmetische Formel $F(x_1, \ldots, x_k, y)$ gibt, so dass für alle $n_1, \ldots, n_k, m \in \mathbb{N}$ folgendes gilt:

$$f(n_1, \ldots, n_k) = m \textbf{ gdw } F(n_1, \ldots, n_k, m) \text{ ist wahr}$$

Funktion	**arithmetische Repräsentation**
$x_1 + x_2$ (Addition)	$y = x_1 + x_2$
$x_1 \cdot x_2$ (Multiplikation)	$y = x_1 * x_2$
x_1 DIV x_2 (ganzzahliger Quotient)	$\exists r < x_2(x_1 = y * x_2 + r)$
x_1 MOD x_2 (ganzzahliger Rest)	$\exists q(x_1 = q * x_2 + y \wedge y < x_2)$

Beobachtung (syntaktischer Zucker):

Arithmetisch repräsentierbare Funktionen können wie Terme eingesetzt werden!

Arithmetische Repräsentation von Programmen

Wir sagen ein WHILE-Programm P mit Variablen $x_0, \ldots, x_k$ hat die arithmetische Repräsentation

$$F_P(x_0, \ldots, x_k, y_0, \ldots, y_k) \ ,$$

wenn für alle $m_0, \ldots, m_k, n_0, \ldots, n_k \in \mathbb{N}$ folgendes gilt:
P gestartet mit den Variablenwerten $m_0, \ldots, m_k$ stoppt nach endlich vielen Schritten mit den Variablenwerten $n_0, \ldots, n_k$ **gdw** $F_P(m_0, \ldots, m_k, n_0, \ldots, n_k)$ wahr ist.

Zentraler Satz:
Zu jedem WHILE-Programm P gibt ist eine arithmetische Repräsentation F_P.

Beweis (strukturelle Induktion)

WHILE-Programm	**arithmetische Repräsentation**
$x_i := x_j + c$	$(y_i = x_j + c) \wedge \bigwedge_{l \neq i}(y_l = x_l)$
$x_i := x_j - c$	$(x_j \geq c \rightarrow y_i + c = x_j) \wedge (x_j < c \rightarrow y_i = 0)$ $\wedge \bigwedge_{l \neq i}(y_l = x_l)$
$Q; R$	$\exists z_0, \ldots, z_k\, (F_Q(x_0, \ldots, x_k, z_0, \ldots, z_k)$ $\wedge F_R(z_0, \ldots, z_k, y_0, \ldots, y_k))$

Der fehlende Induktionsschritt (WHILE-Schleife) ist kompliziert und bedarf eines kurzen Exkurses.

Exkurs: Kompression einer Zahlenfolge

Die Hilfsfunktion (arithmetisch repräsentierbar!)

$$\mathrm{sel}(a, b, i) := a \text{ MOD } (1 + (i + 1)b)$$

nennen wir im Folgenden Selektionsfunktion.

Technischer Hilfssatz

Zu jeder Zahlenfolge $n_0, n_1, \ldots, n_k \in \mathbb{N}$ gibt es zwei natürliche Zahlen $a, b \in \mathbb{N}$, so dass für $i = 0, \ldots, k$:

$$n_i = \mathrm{sel}(a, b, i) \; .$$

Beweisidee: Die Zahlen $b_i := 1 + (i + 1)b$ mit

$$b := s! \text{ und } s := \max\{k, n_0, \ldots, n_k\}$$

sind paarweise teilerfremd. Daher sind die simultanen Kongruenzen

$$a \equiv n_0 \pmod{b_0}, \ldots, a \equiv n_k \pmod{b_k}$$

nach a auflösbar.

Der Fall der WHILE-Schleife

P (mit Variablen $x_0, \ldots, x_k$) habe die Form

$$\text{WHILE } x_i \neq 0 \text{ DO } Q \text{ END}.$$

Wir können induktiv voraussetzen, dass für das WHILE-Programm Q eine arithmetische Repräsentation F_Q existiert.

Es bezeichne

- $z_l(t)$ den Wert der Variablen x_l nach t-maligem Durchlaufen des Schleifenkörpers Q,
- T die Gesamtanzahl der Durchläufe bis zum Erreichen der Abbruchbedingung $x_i = 0$.

Ziel:

Beschreibe durch eine arithmetische Formel, dass P die Anfangswerte $x_0, \ldots, x_k$ in die Werte $y_0, \ldots, y_k$ überführt.

Eine „fast-arithmetische“ Repräsentation

$$\exists z_0, \ldots, z_k : \mathbb{N} \to \mathbb{N},\ \exists T \in \mathbb{N}$$

(Anfangsbedingung$\wedge$Endbedingung$\wedge$Iterationsbedingung $\wedge$Laufzeitbedingung)

Anfangsbedingung: $z_0(0) = x_0 \wedge \cdots \wedge z_k(0) = x_k$

Endbedingung: $z_0(T) = y_0 \wedge \cdots \wedge z_k(T) = y_k$

Iterationsbedingung $\forall t < T\ F_Q(z_0(t), \ldots, z_k(t), z_0(t+1), \ldots, z_k(t+1))$

Laufzeitbedingung $z_i(T) = 0 \wedge \forall t < T\ (z_i(t) > 0)$

Problem:

Es dürfen nur Variable, aber keine Funktionen $z_l : \mathbb{N} \to \mathbb{N}$ „quantifiziert“ werden.

Lösung:

Repräsentiere die Folge $z_l(0), \ldots, z_l(T)$ durch zwei Variable a_l, b_l mit Werten in $\mathbb{N}$ (Kompressionstechnik in Verbindung mit der Selektionsfunktion).

Arithmetische Repräsentation der WHILE-Schleife

$$\exists a_0, b_0 \ldots, a_k, b_k, T$$

(Anfangsbedingung$\wedge$Endbedingung$\wedge$Iterationsbedingung $\wedge$Laufzeitbedingung)

Anfangsbedingung: $\mathrm{sel}(a_0, b_0, 0) = x_0 \wedge \cdots \wedge \mathrm{sel}(a_k, b_k, 0) = x_k$

Endbedingung: $\mathrm{sel}(a_0, b_0, T) = y_0 \wedge \cdots \wedge \mathrm{sel}(a_k, b_k, T) = y_k$

Iterationsbedingung: $\forall t < T \ \exists w_0, w_0' \ldots, w_k, w_k'$

$$(w_0 = \mathrm{sel}(a_0, b_0, t) \wedge \cdots \wedge w_k = \mathrm{sel}(a_k, b_k, t))$$
$$\wedge(w_0' = \mathrm{sel}(a_0, b_0, t+1) \wedge \cdots \wedge w_k' = \mathrm{sel}(a_k, b_k, t+1))$$
$$\wedge F_Q(w_0, \ldots, w_k, w_0', \ldots, w_k')$$

Laufzeitbedingung $\mathrm{sel}(a_i, b_i, T) = 0 \wedge \forall t < T \ (\mathrm{sel}(a_i, b_i, t) > 0)$

Folgerung 1

Satz:

Jede WHILE-berechenbare Funktion ist arithmetisch repräsentierbar.

Beweis:

Sei P ein WHILE-Programm mit Variablen $x_0, \ldots, x_k$ zur Berechnung von $f : \mathbb{N}^n \to \mathbb{N}$ mit $k \geq n$ und F_P die P repräsentierende arithmetische Formel. Dann wird f repräsentiert durch

$$\exists w_1, \ldots, w_k \; F_P(0, \underbrace{x_1, \ldots, x_n}_{Eingabe}, \underbrace{0, \ldots, 0}_{(k-n)-mal}, \underbrace{y}_{Ausgabe}, w_1, \ldots, w_k) \; .$$

Folgerung 2

Satz:

Die Menge WA der wahren arithmetischen Formeln ist unentscheidbar.

Beweis:

Sei $A \subseteq \mathbb{N}$ eine semi-entscheidbare aber unentscheidbare Menge. Dann ist

$$\chi'_A(n) = \begin{cases} 1 & \text{falls } n \in A \\ \text{undefiniert} & \text{sonst} \end{cases}$$

WHILE-berechenbar und daher repräsentierbar durch eine arithmetische Formel $F(x, y)$. Es gilt

$$n \in A \Leftrightarrow \chi'_A(n) = 1 \Leftrightarrow F(n, 1) \text{ wahr} \Leftrightarrow F(n, 1) \in \text{WA} \ .$$

Abbildung $n \mapsto F(n, 1)$ demonstriert, dass A$\leq$WA. Folglich ist WA nicht entscheidbar.

Folgerung 3

Satz:

Die Menge WA der wahren arithmetischen Formeln ist nicht semi-entscheidbar (und somit auch nicht aufzählbar).

(Widerspruchs-)Beweis:

Da für jede arithmetische Formel F entweder F oder $\neg F$ wahr ist, könnten wir WA mit Hilfe eines Akzeptors (der simultan auf F und $\neg F$ angesetzt wird) entscheiden.

Gödel'scher Unvollständigkeitssatz

Jedes (korrekte) Beweissystem für die Menge der arithmetischen Formeln ist notwendigerweise unvollständig (d.h., es bleiben immer wahre arithmetische Formeln übrig, die unbeweisbar sind).

Der (Widerspruchs-)Beweis (der ohne Formalisierung eines Beweissystems auskommt) basiert auf zwei minimalistischen Grundannahmen, die jedes „vernünftige" Beweissystem erfüllt:

1. Die Menge $\mathcal{B}$ der Beweise ist aufzählbar, d.h., es gibt eine berechenbare surjektive Abbildung $f : \mathbb{N} \rightarrow \mathcal{B}$.
2. Aus einem Beweis kann man die durch ihn bewiesene arithmetische Formel „ablesen", d.h., es existiert eine berechenbare Abbildung $g : \mathcal{B} \rightarrow \mathrm{WA}$, die einem Beweis die durch ihn bewiesene Formel zuordnet.

Wäre nun jede Formel aus WA beweisbar, dann erhielten wir eine berechenbare surjektive Abbildung $g \circ f : \mathbb{N} \rightarrow \mathrm{WA}$ und WA wäre aufzählbar (Widerspruch).