Prof. T. Nipkow
K. Kappelmann, J. Rädle, L. Stevens
Technische Universität München
Lehrstuhl für Logik und Verifikation

ABGABE 27.07.2020, 23:59 (Σ 6 Bonuspunkte)

Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt Lösungsskizze 13

Auf diesem Blatt können Sie nochmal 6 Bonuspunkte sammeln $$$

AUFGABE 13.1. (*How lucky I am to have something...*) 0.5 + 0.5 + 0.5P

Bearbeiten Sie die folgenden Aufgaben ohne Verwendung von *LOOP*-Programmen.

(a) Sei $f : \mathbb{N} \to \mathbb{N}$ primitiv-rekursiv und streng monoton wachsend. Zeigen Sie: Die Linksinverse f^{-1} (d.h. $f^{-1}(f(n)) = n$ für alle $n \in \mathbb{N}$) ist primitiv-rekursiv.

(b) Sei $f : \mathbb{N} \to \mathbb{N}$ primitiv-rekursiv und es gelte $n \leq f(n)$. Zeigen Sie, dass $f(a(n, m))$ nicht primitiv-rekursiv ist, wobei $a(\cdot, \cdot)$ die Ackermann-Funktion ist.

(c) Sei $f : \mathbb{N} \to \mathbb{N}$ primitiv-rekursiv. Zeigen oder widerlegen Sie: Die Menge $\{n \in \mathbb{N} \mid n \geq f(n)\}$ ist entscheidbar.

Lösungsskizze

(a) Nach VL ist $f^{-1}(n) \coloneqq \max\{i \leq n \mid f(i) \dot{-} n = 0\}$ primitiv-rekursiv (Folie 271). Da f streng monoton wachsend ist, gilt $n \leq f(n) < f(n+1)$ und somit

$$f^{-1}(f(n)) = \{i \leq f(n) \mid f(i) \dot{-} f(n) = 0\} = \max\{i \leq f(n) \mid f(i) \leq f(n)\} = \max\{i \leq n \mid f(i) \leq f(n)\} = n.$$

(b) Angenommen die Funktion wäre primitiv-rekursiv. Dann wäre auch $g(n) \coloneqq f(a(n, n))$ primitiv-rekursiv Nach Lemma 5.55 gibt es dann $t \in \mathbb{N}$ mit $g(t) < a(t, t) \underbrace{\leq}_{n \leq f(n)} f(a(t, t)) = g(t)$. Widerspruch.

(c) Die Aussage ist wahr. Jede primitiv-rekursive Funktion ist total. Wir können daher bei Eingabe n einfach $f(n)$ berechnen und $n \geq f(n)$ überprüfen.

AUFGABE 13.2. (*... that makes saying goodbye so hard. – Winnie The Pooh*) 1 + 1P

Seien $A, B \subseteq \Sigma^*$ rekursiv aufzählbare Sprachen. Zeigen Sie die folgenden Aussagen, indem Sie geeignete Funktionen angeben, die die jeweilige Sprache auflisten (keine TM-Konstruktionen!):

(a) $\mathsf{L}_1 := A \times B$ ist rekursiv aufzählbar.

(b) $\mathsf{L}_2 := A^*$ ist rekursiv aufzählbar.

Hinweis: Verwenden Sie die Cantorsche Paarungsfunktion.

Lösungsskizze

(a) Der Fall $\mathsf{L}_1 = \emptyset$ ist trivial. Betrachten wir also $\mathsf{L}_1 \neq \emptyset$. Dann muss auch $A \neq \emptyset$ und $B \neq \emptyset$ gelten. Seien f_A und f_B Funktionen, die A bzw. B aufzählen. Dann ist die Funktion $f(n) = (f_A(p_1(n)), f_B(p_2(n)))$ total und berechenbar (da f_A, f_B, p_1, und p_2 total und berechenbar sind) und zählt L_1 auf:
Da $f_A(p_1(n)) \in A$ und $f_B(p_2(n)) \in B$ für alle $n \in \mathbb{N}$, ist $f(n) \in \mathsf{L}_1$.
Sei umgekehrt $x = (y, z) \in \mathsf{L}_1$ beliebig. Dann gilt per Definition $y \in A$ und $z \in B$. Also gibt es $n, m \in \mathbb{N}$, sodass $f_A(n) = y$ und $f_B(m) = z$. Da die Cantorsche Paarfunktion c bijektiv ist, gibt es also auch ein $k \in \mathbb{N}$, sodass $p_1(k) = n$ und $p_2(k) = m$. Somit gilt $f(k) = (y, z) = x$; f zählt also alle Elemente von L_1 auf.

(b) Sei f die Funktion die A aufzählt. Wir zerlegen zuerst ein gegebenes $n \in \mathbb{N}$ in das Paar $(z, k) = (p_1(n), p_2(n))$. Wir verwenden k um zu entscheiden, in wie viele Teile wir ein Wort zerlegen und z um zu entscheiden, was diese Teilwörter sein sollen. Wir definieren also die Funktion $f''(z, k) \coloneqq f(d_0(z)) \dots f(d_{k-1}(z))$ und $f'(n) \coloneqq f''(p_1(n), p_2(n))$, die total und berechenbar ist, und L_2 aufzählt:
Da $f(\mathbb{N}) = A$, gilt für alle i, dass $f(d_i(z)) \in A$ und somit $f'(n) \in A^*$.
Sei $w \in A^*$ beliebig. Dann gibt es $w_1, \dots, w_k \in A$ mit $w_1 \dots w_k = w$. Da A von f aufgezählt wird, gibt es $n_1, \dots, n_k$ mit $f(n_i) = w_i$. Da die Cantorsche Paarfunktion bijektiv ist, gibt es $x \in \mathbb{N}$ mit $x = \langle \langle n_1, \dots, n_k \rangle, k \rangle$ und dann

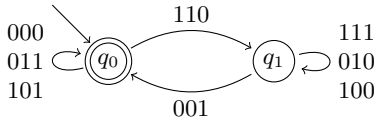
$$f'(x) = f(d_0(\langle n_1, \dots, n_k \rangle)) \cdots f(d_{k-1}(\langle n_1, \dots, n_k \rangle)) = w_1 \cdots w_k = w$$

.

AUFGABE 13.3. (*Increment the homework points*) 1 + 0.5 + 0.5 + 0.5P

In dieser Aufgabe zeigen wir, dass die Lösungsmenge für einen Teil der logischen Formeln über den natürlichen Zahlen regulär ist. Wir können die Erfüllbarkeit dieser Formeln also entscheiden, indem wir die Sprache auf Leerheit prüfen. Dabei stellen wir die Zahlen binär da, wobei das erste Bit das niederwertigste Bit ist. Wir konstruieren jeweils einen Automaten, der die Eingabe akzeptiert, wenn die logische Formel wahr ist. Wenn die Eingabe aus mehreren Zahlen besteht, dann werden dieses stets als Tupel eingelesen. Die Darstellung der Zahlen ist hierbei nicht eindeutig, da am Ende beliebig viele Nullen folgen können, ohne den Wert der Zahl zu verändern. Für die

Formel $x + y = z$ konstruieren wir den folgenden NFA, wobei bie jedem Übergang das erste Bit zu x, das zweite zu y und das dritte zu z gehört:

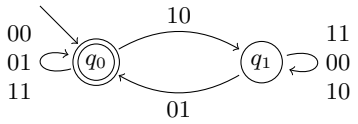


Gehen Sie folgendermaßen vor:

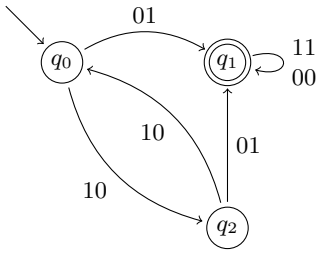
(a) Geben Sie einen NFA für die Formel $\exists y.\ x + y = z$ an.
(b) Geben Sie einen NFA für die Formel $x \leq y$ an.
(c) Geben Sie einen NFA für die Formel $x + 1 = y$ an.
(d) Begründen Sie, dass man auch eine Verundung $\varphi_1 \wedge \varphi_2$ von zwei Formeln durch einen NFA darstellen kann.

Lösungsskizze

(a) Projeziere die zweite Komponente weg.



(b) Die Formel $\exists y.\ x + y = z$ ist bereits äquivalent zu $x \leq z$, da wir nur natürliche Zahlen betrachten.
(c)



(d) Sei N_1 der NFA für φ_1 und F_1 die Menge der freien Variablen in φ_1. Analog definieren wir N_2 und F_2. Wir verwenden die Produktkonstruktion, um den Schnitt von N_1 und N_2 zu bilden. Davor müssen wir allerdings die Übergänge von N_1 (bzw. N_2) so erweitern, dass der Automat alle Werte für freie Variablen in $F_2 \setminus F_1$ (bzw. $F_1 \setminus F_2$) akzeptiert.

Hinweis: Diese eingeschränkten logischen Formeln werden Presburger-Arithmetik genannt. Indem man Automaten für alle logischen und arithmetischen Operatoren angibt, kann man zeigen, dass Presburger-Arithmetik im Gegensatz zur Arithmetik mit Multiplikation entscheidbar ist.

> The more I think about language, the more it amazes me that people ever understand each other at all.
>
> — Kurt Gödel

Bonus: Eine wunderschöne Antwort zu "What is the enlightenment I'm supposed to attain after studying finite automata?"

Vielen Dank für die Mitarbeit während dieses etwas anderem Semesters! Wir wünschen viel Erfolg bei der Klausur und Freude beim zukünftigen THEOretisieren, Beweisen und Philosphieren.