

Grundbegriffe der Informatik — Aufgabenblatt 6

Lösungsvorschläge

Tutorium Nr.:

Tutor*in:

Matr.nr. 1:

Nach-,Vorname 1:

,

Matr.nr. 2:

Nach-,Vorname 2:

,

Ausgabe:

Freitag, 2.12.2022, 14:30 Uhr

Abgabe:

Freitag, 9.12.2022, 12:30 Uhr

Online, oder in einem Briefkasten mit der Aufschrift GBI
im UG des Info-Gebäudes (50.34)

Lösungen werden nur korrigiert, wenn sie

- handschriftlich erstellt sind (Tablet-Ausdruck erlaubt) und
- mit dieser Seite als Deckblatt
- in der oberen **linken** Ecke zusammengeheftet **rechtzeitig** abgegeben werden.

Abgaberegeln für Teilnehmer der Tutorien mit Online-Abgabe:

- handschriftlich erstellt (Scans und lesbare Fotos akzeptiert)
- **rechtzeitig**, mit diesem Deckblatt in **genau einer** PDF-Datei
- in ILIAS unter "Tutorien" im Ordner des richtigen Tutoriums abgeben.

Von Tutor*in auszufüllen: erreichte Punkte

Blatt 6:

 / 19

Blätter 1 – 6, Stud. 1:

 / 124

Blätter 1 – 6, Stud. 2:

 / 124

Aufgabe 6.1 (1 + 1 + 1 + 1 = 4 Punkte)

Sei $A = \{a, b, c\}$ ein Alphabet.

- Zeigen oder widerlegen Sie: Es gibt einen Homomorphismus $f : A^* \rightarrow Z_2^*$ mit $f(aaa) = 011110$.
- Geben Sie einen injektiven, aber nicht präfixfreien Homomorphismus $h : A^* \rightarrow Z_2^*$ an.

Wir betrachten für eine Abbildung $g : A^* \rightarrow Z_2^*$ die Gleichung

$$g(aca) = g(bbca) . \quad (1)$$

- Geben Sie die vollständige Definition eines ε -freien Homomorphismus $g : A^* \rightarrow Z_2^*$ an, sodass die Gleichung (1) erfüllt ist.
- Sei g nun ein beliebiger ε -freier Homomorphismus, für den die Gleichung (1) gilt.
 - Sei $n \in \mathbb{N}_+$ die Länge von $g(b)$. Was lässt sich damit über $|g(a)|$ aussagen?
 - Sei zusätzlich $m \in \mathbb{N}_+$ die Länge von $g(c)$. Geben Sie die Länge des Wortes $g(bacca)$ in Abhängigkeit von n und m an.

Lösung 6.1

- f kann kein Homomorphismus sein.

Beweis durch Widerspruch:

Angenommen f wäre ein Homomorphismus, dann müsste gelten:

$$f(aaa) = f(a) \cdot f(a) \cdot f(a) = 011110 = w \quad (2)$$

Dann gilt $|f(a)| = \frac{1}{3}|w| = 2$. Folglich müsste gelten $0 = w(0) = f(a)(0) = w(2) = 1$. Diese Gleichheit ist aber offensichtlich verletzt.

- Der Homomorphismus $h : A^* \rightarrow Z_2^*$ mit

$$h(a) = 0 \quad h(b) = 01 \quad h(c) = 11$$

ist nicht präfixfrei, da $h(a) = 0$ Präfix von $h(b) = 01$ ist. Da h suffixfrei ist, ist h injektiv.

- Ein solcher Homomorphismus g ist beispielweise gegeben durch

$$g(a) = 11 \quad g(b) = 1 \quad g(c) = 0 . \quad (3)$$

- Es gilt

$$g(aca) = g(bbca) \quad (4)$$

$$\text{gdw. } g(a) \cdot g(ca) = g(bb) \cdot g(ca) . \quad (5)$$

Damit die Gleichung erfüllt ist, müssen die Präfixe und Suffixe auf der linken und rechten Seite übereinstimmen. Es folgt also, dass $g(a) = g(bb)$ sein muss. Damit gilt: $|g(a)| = |g(bb)| = |g(b) \cdot g(b)| = 2|g(b)|$.

- Nach Teilaufgabe (i) gilt $g(a) = g(bb)$ es folgt also für $w = bacca$:

$$|w| = |g(bacca)| \quad (6)$$

$$= |g(b) \cdot g(a) \cdot g(c) \cdot g(c) \cdot g(a)| \quad (7)$$

$$= |g(b) \cdot g(bb) \cdot g(c) \cdot g(c) \cdot g(bb)| \quad (8)$$

$$= |g(b)| + 2 \cdot |g(b)| + |g(c)| + |g(c)| + 2 \cdot |g(b)| \quad (9)$$

$$= 5 \cdot |g(b)| + 2 \cdot |g(c)| \quad (10)$$

$$= 5n + 2m \quad (11)$$

Aufgabe 6.2 (3 + 2 + 5 = 10 Punkte)

Diese Aufgabe befasst sich mit dem Begriff der Linksinversen.

a) Sei $A = \{a, e, m, t\}$ und $h : A \rightarrow Z_2^*$ gegeben durch

$$h(a) = 000 \quad h(e) = 1 \quad h(m) = 001 \quad h(t) = 01.$$

- i) Geben Sie eine Linksinverse $g : Z_2^* \rightarrow A$ zu h an.
 - ii) Geben Sie mindestens 100 000 weitere verschiedene Linksinverse $g_n : Z_2^* \rightarrow A$ zu h an (natürlich ohne sie einzeln aufzuzählen).
- b) Seien nun $f : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}, x \mapsto x^2$ und $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}, x \mapsto \sqrt{x}$ gegeben.
- i) Ist g eine Linksinverse zu f ?
 - ii) Ist g eine Rechtsinverse zu f ?

Begründen Sie jeweils kurz.

- c) In der Vorlesung wurde – ohne Beweis – vorgestellt, dass eine Abbildung „ f genau dann eine Linksinverse hat, wenn f injektiv ist“. Eine Aussage in einer Universitäts-Vorlesung ohne Beweis! Das können Sie natürlich nicht so stehen lassen. Seien also A, B zwei beliebige nicht-leere Mengen und $f : A \rightarrow B$ eine Abbildung. Wir wollen zeigen: f hat genau dann eine Linksinverse, wenn f injektiv ist. Dazu sollen Sie die Implikation in beide Richtungen zeigen:

- i) Beweisen Sie die Implikation: Wenn f injektiv ist, dann hat f eine Linksinverse. *Hinweis:* Sie können dazu eine Relation g angeben und beweisen, dass g eine Funktion, sowie linksinvers zu f ist.
- ii) Beweisen Sie die Implikation: Wenn f eine Linksinverse hat, dann ist f injektiv. *Hinweis:* Sie können dazu die Kontraposition der Implikation beweisen.

Lösung 6.2

a) i) Jede Linksinverse g muss erfüllen, dass

$$g(000) = a \quad g(1) = e \quad g(001) = m \quad g(01) = t \quad (12)$$

Darüber hinaus ist eine Linksinverse auch eine Abbildung, g muss also insbesondere linkstotal sein. Daher muss auch ein Funktionswert von g für die übrigen $w \in Z_2^*$ existieren. Wir wählen dafür einfach

$$g(w) = a$$

für $w \in Z_2^* \setminus \{000, 1, 001, 01\}$.

- ii) Wir können beliebig viele verschiedene Linksinverse angeben, indem wir den $w \in Z_2^* \setminus \{000, 1, 001, 01\}$, die keinen weiteren Bedingungen unterliegen, als dass ein Funktionswert für sie existieren muss, auf verschiedene Funktionswerte abbilden. Für beliebige $n \in \mathbb{N}_+$ mit $n \geq 3$ sei also g_n definiert durch

$$g_n(000) = a \quad g_n(1) = e \quad g_n(001) = m \quad g_n(01) = t \quad (13)$$

$$g_n(0^n 1) = a \quad (14)$$

$$g_n(w) = m \quad \forall w \in Z_2^* \setminus \{000, 1, 001, 01, 0^n 1\} \quad (15)$$

Für verschiedene $n \in N_+$ mit $n \geq 3$ sind die resultierenden g_n offensichtlich verschieden, da von den $w \in Z_2^* \setminus \{000, 1, 001, 01\}$ nur $0^n 1$ auf **a** abgebildet wird und alle anderen w auf **m**. Da es unendlich viele $n \in N_+$ mit $n \geq 3$ und damit unendlich viele verschiedene g_n gibt, gibt es auch mindestens 100000 solcher g_n .

- b) i) Für eine Linksinverse g zu f muss gelten, dass $(g \circ f)(x) = x, \forall x \in \mathbb{R}$. Es ist aber $(g \circ f)(-5) = g(f(-5)) = \sqrt{(-5)^2} = \sqrt{25} = 5 \neq -5$. Damit ist g keine Linksinverse zu f .
- ii) g ist eine Abbildung und es gilt $\forall x \in \mathbb{R}_{\geq 0} : (f \circ g)(x) = (\sqrt{x})^2 = x$, damit ist g eine Linksinverse für f .
- c) Um die Äquivalenz zu zeigen, beweisen wir die Implikation in beide Richtungen.
- i) f injektiv $\Rightarrow f$ hat Linksinverse.

Wir definieren die Relation $g \subseteq B \times A$ durch

$$g = \{(f(a), a) \mid \forall a \in A\} \cup \{(b, x) \in B \times A \mid b \notin f(A)\} \quad (16)$$

für ein beliebiges aber festes Element $x \in A$.

Linkstotal Wir zeigen: g ist linkstotal.

Es muss also gelten: $\forall b \in B : \exists a \in A : (b, a) \in g$

Sei also $b \in B$ beliebig. Wir können zwei Fälle unterscheiden:

i. Fall 1: $b \in f(A)$

Dann existiert ein $a \in A$ sodass $b = f(a)$ und damit $(f(a), a) = (b, a) \in g$.

ii. Fall 2: $b \in B \setminus f(A)$

Dann ist $b \notin f(A)$ und $(b, x) \in g$.

Folglich ist g linkstotal.

Rechtseindeutig Wir zeigen: g ist rechtseindeutig.

Es muss also gelten: $\forall b \in B : \forall a, a' \in A : (b, a) \in g$ wenn $(b, a') \in g$ dann $a = a'$.

Seien also $(b, a) \in g$ und $(b, a') \in g$. Wir unterscheiden wieder zwei Fälle:

i. Fall 1: $b \in f(A)$

Nach Definition von g gibt es dann $a, a' \in A$, sodass $(b, a) = (f(a), a)$ und $(b, a') = (f(a'), a')$ und damit $f(a) = b = f(a')$. Mit der Injektivität von f gilt dann $f(a) = f(a') \Rightarrow a = a'$.

ii. Fall 2: $b \in B \setminus f(A)$

Dann ist $b \notin f(A)$ und für solche b ist nur $(b, x) \in g$. Damit muss gelten $a = x = a'$.

Folglich ist g rechtseindeutig.

Zusammen mit der Linkstotalität ist g damit auch eine Abbildung und wir können schreiben:

$$g(b) = \begin{cases} a, & \text{falls } \exists a \in A : f(a) = b \\ x, & \text{sonst} \end{cases} \quad (17)$$

Linksinvers Wir zeigen g ist linksinvers zu f
 Es muss also gelten: $\forall a \in A : (g \circ f)(a) = g(f(a)) = a$.
 Für $b = f(a)$ gilt, dass $g(b) = a$ und damit $g(f(a)) = a$.

ii) f hat Linksinverse $\Rightarrow f$ ist injektiv

Wir wenden Kontraposition an und beweisen stattdessen: f ist nicht injektiv $\Rightarrow f$ hat keine Linksinverse .

Sei also f nicht injektiv. Dann gibt es $a, a' \in A$ mit $a \neq a'$, sodass $f(a) = f(a')$.
 Für eine Linksinverse muss gelten, dass $(g \circ f)(x) = x \forall x \in A$. Damit muss auch für a, a' gelten, dass

$$(g \circ f)(a) = g(f(a)) = a \text{ und}$$

$$(g \circ f)(a') = g(f(a')) = a' .$$

Nach Annahme ist aber $f(a) = f(a')$ und $a \neq a'$. Daher gibt es keine solche rechtseindeutige Relation und damit auch keine solche Abbildung.

Aufgabe 6.3 (3 + 2 = 5 Punkte)

Wir wollen uns mit Lese- und Schreibe-Operation im mathematischen Speichermodell aus der Vorlesung beschäftigen. Dazu seien wieder Adr und Val die Menge der Adressen, bzw. die Menge der an Adressen speichbaren Werte. Verwenden Sie die in der Vorlesung vorgestellten Funktionen memread und memwrite , um lesend und schreibend auf Speicher zu operieren.

- a) Sei $A = \{a, b, c, d, e, f, g\}$ und für diese Teilaufgabe $\text{Val} = \mathbb{Z}_8$; es werden also an jeder Speicherstelle drei Bit gespeichert, die als natürliche Zahl interpretiert werden. Wir suchen nach einer Funktion

$$\text{to_string}: \text{Val}^{\text{Adr}} \times \text{Adr} \rightarrow A^*$$

$$(m, a) \mapsto \text{to_string}(m, a) ,$$

die aus einem Speicher $m \in \text{Val}^{\text{Adr}}$ eine Zeichenkette (d.h. ein Wort) in A^* ausliest, das bei einer Adresse a beginnt, in aufsteigender Adressreihenfolge fortgeht (solange möglich) und endet, wenn der Nullwert $0 \in \mathbb{Z}_8$ im Speicher angetroffen wird.¹ Die Zuordnung von Speicherinhalten zu Symbolen in A erfolgt mittels folgender Funktion $d: \text{Val} \setminus \{0\} \rightarrow A$:

$$d = \{(1, a), (2, b), (3, c), (4, d), (5, e), (6, f), (7, g)\}$$

Beispiel: Betrachten wir einen Speicherausschnitt eines Speichers $m \in \text{Val}^{\text{Adr}}$ für einige Speicherstellen a ab 531. Die zweite Zeile enthält für jede Adresse a den Wert in m an dieser Stelle und die letzte Zeile zeigt den als Zeichen in A decodierte Speicherinhalt an der Stelle a . Gemäß der Anforderungen an to_string gilt hier: $\text{to_string}(m, 532) = \text{deadbeef}$

a	531	532	533	534	535	536	537	538	539	540
$\text{memread}(m, a)$	3	4	5	1	4	2	5	5	6	0
$d(\text{memread}(m, a))$	c	d	e	a	d	b	e	e	f	-

¹Eine solche „0-Terminierung“ ist ein gängiges Verfahren, um das Ende von Zeichenketten zu markieren und findet z. B. in der Programmiersprache C Anwendung.

- i) Geben Sie eine Definition der Funktion to_string an für $Adr = \mathbb{Z}_{1024}$, also für einen endlichen Speicher mit 1024 Einträgen. **Hinweis:** Beachten Sie insbesondere, was an den „Grenzen“ des Definitionsbereichs von Adr gelten soll.
- ii) Das mathematische Speichermodell kann auch unendlichen Speicher modellieren. Welches Problem tritt auf, wenn Sie die Definition aus i) auf die Situation $Adr = \mathbb{N}$ übertragen, in der es keine Speicherobergrenze gibt? Schlagen Sie eine Lösung vor, damit die Definition von to_string sinnvoll bleibt.
- b) Wir kennen schon eine Funktion, die eine Speicherstelle beschreibt, das wollen wir nun verallgemeinern.
Sei $Adr = \mathbb{N}$. Geben Sie eine Definition für die Funktion

$$memset: Val^{Adr} \times Adr \times \mathbb{N}_0 \times Val \rightarrow Val^{Adr}$$

$$(m, a, n, v) \rightarrow m'$$

an, die in einem Speicher m den Wert v an n aufeinanderfolgende Speicherstellen ab der Adresse a schreibt (und nur diesen Bereich verändert) und den so veränderten Speicher m' als Ergebnis liefert.

Lösung 6.3

a) i)

$$to_string(m, a) = \begin{cases} \varepsilon, & \text{falls } memread(m, a) = 0 \\ d(memread(m, a)), & \text{falls } memread(m, a) \neq 0 \text{ und } a = 1023 \\ d(memread(m, a)) \cdot to_string(m, a + 1), & \text{sonst} \end{cases}$$

Es gilt insbesondere, dass to_string nach Aufgabenstellung solange wie möglich in aufsteigender Adressreihenfolge fortgeht. Nachdem das Ende des Adressraumes erreicht ist, darf also nicht bei der kleinsten Adresse weitergelesen werden.

- ii) Als Funktion muss to_string jedem Eingabepaar $m, a \in Val^{Adr} \times Adr$ einen Wert zuweisen. Dafür muss to_string notwendigerweise für jedes Eingabepaar terminieren. Es muss also vorausgesetzt werden, dass nach a mindestens eine 0 im Speicher steht, damit die Rekursion nicht ewig fortläuft.
- b)

$$memset(m, a, n, v) = \begin{cases} m & , \text{ falls } n = 0 \\ memset(memwrite(m, a, v), a + 1, n - 1, v) & , \text{ falls } n > 0 \end{cases}$$