Grundbegriffe der Informatik — Aufgabenblatt 6

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

a) Zeigen oder widerlegen Sie: Es gibt einen Homomorphismus $f : A^* \to Z_2^*$ mit $f(\mathtt{aaa}) = \mathtt{011110}$.

b) Geben Sie einen injektiven, aber nicht präfixfreien Homomorphismus $h : A^* \to Z_2^*$ an.

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

$$g(\mathtt{aca}) = g(\mathtt{bbca}) \,. \tag{1}$$

c) Geben Sie die vollständige Definition eines ε-freien Homomorphismus $g : A^* \to Z_2^*$ an, sodass die Gleichung (1) erfüllt ist.

d) Sei g nun ein beliebiger ε-freier Homomorphismus, für den die Gleichung (1) gilt.

   (a) Sei $n \in \mathbb{N}_+$ die Länge von $g(\mathtt{b})$. Was lässt sich damit über $|g(\mathtt{a})|$ aussagen?

   (b) Sei zusätzlich $m \in \mathbb{N}_+$ die Länge von $g(\mathtt{c})$. Geben Sie die Länge des Wortes $g(\mathtt{bacca})$ in Abhängigkeit von n und m an.

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

Diese Aufgabe befasst sich mit dem Begriff der Linksinversen.

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

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

   i) Geben Sie eine Linksinverse $g : Z_2^* \to A$ zu h an.

   ii) Geben Sie mindestens 100 000 weitere verschiedene Linksinverse $g_n : Z_2^* \to A$ zu h an (natürlich ohne sie einzeln aufzuzählen).

b) Seien nun $f : \mathbb{R} \to \mathbb{R}_{\geq 0}, x \mapsto x^2$ und $g : \mathbb{R}_{\geq 0} \to \mathbb{R}, x \mapsto \sqrt{x}$ 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 $f : A \to B$ 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.

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 $A = \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}, \texttt{g}\}$ 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

$$\begin{aligned} \mathit{to\_string}\colon \text{Val}^{\text{Adr}} \times \text{Adr} &\to A^* \\ (m, a) &\mapsto \mathit{to\_string}(m, a) \ , \end{aligned}$$

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 $d : \text{Val} \setminus \{0\} \to A$:

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

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: $\mathit{to\_string}(m, 532) = \texttt{deadbeef}$

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

i) Geben Sie eine Definition der Funktion *to_string* an für $\text{Adr} = \mathbb{Z}_{1024}$, 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 $\text{Adr} = \mathbb{N}$ 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 $\text{Adr} = \mathbb{N}$. Geben Sie eine Definition für die Funktion

$$\begin{aligned} \mathit{memset}\colon \text{Val}^{\text{Adr}} \times \text{Adr} \times \mathbb{N}_0 \times \text{Val} &\to \text{Val}^{\text{Adr}} \\ (m, a, n, v) &\to m' \end{aligned}$$

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.

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