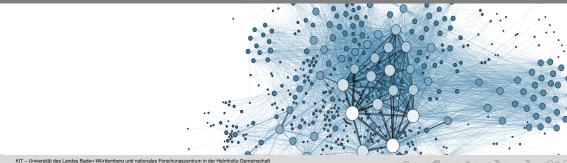




Grundbegriffe der Informatik **Tutorium 33**

Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de Lukas Bach, lukas.bach@student.kit.edu | 24.11.2016



Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Sei Sem die Menge der Bedeutungen.

Homomorphismen

Huffman Codierung

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Sei Sem die Menge der Bedeutungen. Ferner seien A und B Alphabete

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Sei *Sem* die Menge der Bedeutungen. Ferner seien *A* und *B* Alphabete und $L_A \subseteq A^*$ und $L_B \subseteq B^*$.

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Sei *Sem* die Menge der Bedeutungen. Ferner seien *A* und *B* Alphabete und $L_A \subseteq A^*$ und $L_B \subseteq B^*$.

Homomorphismen

Weiter sei $sem_A : L_A \rightarrow Sem$

Huffman Codierung

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Sei *Sem* die Menge der Bedeutungen. Ferner seien *A* und *B* Alphabete und $L_A \subset A^*$ und $L_B \subset B^*$.

Homomorphismen

Weiter sei $sem_A : L_A \rightarrow Sem$ und $sem_B : L_B \rightarrow Sem$

Huffman Codierun

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Sei *Sem* die Menge der Bedeutungen. Ferner seien *A* und *B* Alphabete und $L_A \subset A^*$ und $L_B \subset B^*$.

Homomorphismen

Weiter sei $sem_A: L_A \rightarrow Sem$ und $sem_B: L_B \rightarrow Sem$

Huffman Codierung

Dann heißt $f: L_A \rightarrow L_B$ Übersetzung

Übersetzungen



Maximilian Staab,

Speicher

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen Sei Sem die Menge der Bedeutungen. Ferner seien A und B Alphabete

Homomorphismen und $L_A \subseteq A^*$ und $L_B \subseteq B^*$.

Weiter sei $sem_A : L_A \rightarrow Sem$ und $sem_B : L_B \rightarrow Sem$

Dann heißt $f: L_A \to L_B$ Übersetzung , wenn gilt: für jedes $w \in L_A$ gilt

 $sem_A(w) = sem_B(f(w)).$

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Homomorphismen

Huffman Codierung

Traininair Godiore

Speicher

Sei *Sem* die Menge der Bedeutungen. Ferner seien A und B Alphabete und $L_A \subseteq A^*$ und $L_B \subseteq B^*$.

Weiter sei $sem_A : L_A \rightarrow Sem$ und $sem_B : L_B \rightarrow Sem$

Dann heißt $f: L_A \to L_B$ Übersetzung , wenn gilt: für jedes $w \in L_A$ gilt $sem_A(w) = sem_B(f(w))$.

Bedeutungserhaltende Abbildungen von Wörtern auf Wörter

Beispiel

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Sei *Sem* die Menge der Bedeutungen. Ferner seien *A* und *B* Alphabete und $L_A \subset A^*$ und $L_B \subset B^*$.

Homomorphismen

Weiter sei $sem_A : L_A \rightarrow Sem$ und $sem_B : L_B \rightarrow Sem$

Huffman Codierung

Dann heißt $f: L_A \to L_B$ Übersetzung , wenn gilt: für jedes $w \in L_A$ gilt $sem_A(w) = sem_B(f(w))$.

Speicher

Bedeutungserhaltende Abbildungen von Wörtern auf Wörter

Beispiel

Betrachte $\mathit{Trans}_{2,16}: \mathbb{Z}^*_{16} \to \mathbb{Z}^*_2$ mit $\mathit{Trans}_{2,16}(w) = \mathit{Repr}_2(\mathit{Num}_{16}(w))$

Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Definition der Semantikabbildung

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Sei *Sem* die Menge der Bedeutungen. Ferner seien *A* und *B* Alphabete

und $L_A \subseteq A^*$ und $L_B \subseteq B^*$.

Weiter sei $sem_A: L_A \rightarrow Sem$ und $sem_B: L_B \rightarrow Sem$

Dann heißt $f: L_A \to L_B$ Übersetzung , wenn gilt: für jedes $w \in L_A$ gilt $sem_A(w) = sem_B(f(w))$.

Bedeutungserhaltende Abbildungen von Wörtern auf Wörter

Beispiel

Betrachte $\textit{Trans}_{2,16}: \mathbb{Z}^*_{16} \to \mathbb{Z}^*_2$ mit $\textit{Trans}_{2,16}(w) = \textit{Repr}_2(\textit{Num}_{16}(w))$

• $Trans_{2,16}(A3) = Repr_2(Num_{16}(A3)) = Repr_2(163) = 10100011$

Wozu Übersetzungen



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Wozu Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

■ Lesbarkeit (vergleiche *DF*₁₆ mit 11011111₂)

Huffman Codierung

Wozu Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Lesbarkeit (vergleiche DF₁₆ mit 11011111₂)

Huffman Codierung

Verschlüsselung

Wozu Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Lesbarkeit (vergleiche DF₁₆ mit 11011111₂)

Verschlüsselung

Kompression (Informationen platzsparend aufschreiben)

Wozu Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

- Lesbarkeit (vergleiche *DF*₁₆ mit 11011111₂)
- Verschlüsselung
- Kompression (Informationen platzsparend aufschreiben)
- lacktriangle Kontextabhängige Semantiken (Deutsch ightarrow Englisch)

Wozu Übersetzungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

- Lesbarkeit (vergleiche DF₁₆ mit 11011111₂)
- Verschlüsselung
- Kompression (Informationen platzsparend aufschreiben)
- lacktriangle Kontextabhängige Semantiken (Deutsch ightarrow Englisch)
- Fehlererkennung

Codierungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

lukas.bachestudent.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierung

Codierungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

■ Codewort *f*(*w*)

Homomorphismen

Huffman Codierung

Codierungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

■ Codewort f(w) einer Codierung $f: L_A \rightarrow L_B$

Codierungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

■ Codewort f(w) einer Codierung $f: L_A \rightarrow L_B$

• Code: $\{f(w)|w\in L_A\}=f(L_A)$

Codierungen



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach.

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierung

• Codewort f(w) einer Codierung $f: L_A \to L_B$

• Code: $\{f(w)|w\in L_A\}=f(L_A)$

Codierung: Injektive Übersetzung

Codierungen



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach.

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierung

- Codewort f(w) einer Codierung $f: L_A \to L_B$
- Code: $\{f(w)|w\in L_A\}=f(L_A)$
- Codierung: Injektive Übersetzung
 - Ich komme immer eindeutig von einem Codewort f(w) zu w zurück

Codierungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

■ Codewort f(w) einer Codierung $f: L_A \rightarrow L_B$

• Code: $\{f(w)|w\in L_A\}=f(L_A)$

Codierung: Injektive Übersetzung

■ Ich komme immer eindeutig von einem Codewort f(w) zu w zurück

Bemerkung

Codierungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierun

Speicher

- Codewort f(w) einer Codierung $f: L_A \to L_B$
 - Code: $\{f(w)|w\in L_A\}=f(L_A)$
 - Codierung: Injektive Übersetzung
 - Ich komme immer eindeutig von einem Codewort f(w) zu w zurück

Bemerkung

• Was ist, wenn L_A unendlich ist (man kann nicht alle Möglichkeiten aufzählen)

Codierungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Definitionen

Übersetzungen

Homomorphismen

Huffman Codierun

Speicher

- Codewort f(w) einer Codierung $f: L_A \rightarrow L_B$
- Code: $\{f(w)|w\in L_A\}=f(L_A)$
- Codierung: Injektive Übersetzung
 - Ich komme immer eindeutig von einem Codewort f(w) zu w zurück

Bemerkung

- Was ist, wenn L_A unendlich ist (man kann nicht alle Möglichkeiten aufzählen)
- Auswege: Homomorphismen, Block-Codierungen

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Homomorphismen

Huffman Codierung

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Seien A, B Alphabete.

Homomorphismen

Huffman Codierung

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Seien A, B Alphabete. Dann ist $h: A^* \to B^*$

Homomorphismen

Huffman Codierung

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Seien A, B Alphabete. Dann ist $h: A^* \to B^*$ ein Homomorphismus

Homomorphismen

Huffman Codierung

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas, bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Homomorphismen

Seien A, B Alphabete. Dann ist $h: A^* \to B^*$ ein Homomorphismus, falls für alle $w_1, w_2 \in A^*$ gilt:

Huffman Codierung

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Seien A, B Alphabete. Dann ist $h: A^* \to B^*$ ein Homomorphismus, falls für alle $w_1, w_2 \in A^*$ gilt:

Homomorphismen

Huffman Codierung

$$h(w_1w_2) = h(w_1)h(w_2)$$

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Homomorphismen

Seien A, B Alphabete. Dann ist $h: A^* \to B^*$ ein Homomorphismus, falls für alle $w_1, w_2 \in A^*$ gilt:

Huffman Codierur

$$h(w_1w_2) = h(w_1)h(w_2)$$

Speicher

Ein Homomorphismus ist Abbildung, die mit Konkatenation verträglich ist

Homomorphismen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Definition von Homomorphismen

Übersetzungen

Homomorphismen

Seien A, B Alphabete. Dann ist $h: A^* \to B^*$ ein Homomorphismus, falls für alle $w_1, w_2 \in A^*$ gilt:

Huffman Codierur

$$h(w_1w_2) = h(w_1)h(w_2)$$

- Ein Homomorphismus ist Abbildung, die mit Konkatenation verträglich ist
- Homomorphismus ist ε -frei, wenn für jedes $x \in A$: $h(x) \neq \varepsilon$

Homomorphismen

alle $w_1, w_2 \in A^*$ gilt:



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach, lukas bach@student.kit.edu

ach@student.kit.

Definition von Homomorphismen

Übersetzungen

Homomorphismen

Huffman Codierur

Hullillall Gouletul

Seien A, B Alphabete. Dann ist $h: A^* \to B^*$ ein Homomorphismus, falls für

$$h(w_1w_2) = h(w_1)h(w_2)$$

- Ein Homomorphismus ist Abbildung, die mit Konkatenation verträglich ist
- Homomorphismus ist ε -frei, wenn für jedes $x \in A$: $h(x) \neq \varepsilon$
- lacktriangle Homomorphismen lassen das leere Wort unverändert, also $h(\varepsilon)=\varepsilon$

Sei h ein Homomorphismus.

Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas.bach@student.kit.e

Übung zu Homomorphismer

Übersetzungen

Homomorphismen

Huffman Codierung

Sei h ein Homomorphismus.

Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas.bach@student.kit.e

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

Übersetzungen

Homomorphismen

Huffman Codierung

Sei h ein Homomorphismus.

Maximilian Staab,
maximilian.staab@fsmi.ur
Lukas Bach,
lukas bach@student kit.e

Übung zu Homomorphismen

1 h(a) -

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

 $\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$

Übersetzungen

Homomorphismen

Huffman Codierung

Sei *h* ein Homomorphismus.

Maximilian Staab. maximilian.staab@fsmi.ur Lukas Bach. lukas bach@student kit

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

$$\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$$

2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?

Übersetzungen

Homomorphismen

Sei *h* ein Homomorphismus.

Maximilian Staah maximilian.staab@fsmi.ur Lukas Bach. lukas bach@student kit

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

$$\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$$

2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?

Übersetzungen

Homomorphismen

 \rightarrow aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$

Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas.bach@student.kit.e

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Sei *h* ein Homomorphismus.

Übung zu Homomorphismen

- 1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?
- $\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$
- 2. Sei h(a)=01, h(b)=11 und $h(c)=\varepsilon$. Nun sei h(w)=011101. Was war w?
- → aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$ ε -Freiheit hat also die Eindeutigkeit zerstört!

Sei *h* ein Homomorphismus.

Maximilian Staab maximilian.staab@fsmi.ur Lukas Bach. lukas.bach@student.kit.

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

$$\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$$

2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?

 \rightarrow aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$ ε -Freiheit hat also die Eindeutigkeit zerstört!

3. Kann h aus 2 eine Codierung sein?

Übersetzungen

Homomorphismen

Sei *h* ein Homomorphismus.

Maximilian Staab maximilian.staab@fsmi.ur Lukas Bach. lukas.bach@student.kit.

Übersetzungen

Homomorphismen

Speicher

Übung zu Homomorphismen

- 1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?
- $\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$
- 2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?
- \rightarrow aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$ ε -Freiheit hat also die Eindeutigkeit zerstört!
 - 3. Kann h aus 2 eine Codierung sein?
- → Nein, da nicht injektiv!

Sei *h* ein Homomorphismus.

Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas.bach@student.kit.e

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

$$\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$$

- 2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?
- → aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$ ε -Freiheit hat also die Eindeutigkeit zerstört!
 - 3. Kann h aus 2 eine Codierung sein?
- → Nein, da nicht injektiv!
- 4. Warum will man ε -freie Homomorphismen?

Übersetzungen

Homomorphismen

Huffman Codierung



Sei *h* ein Homomorphismus.

Maximilian Staab maximilian.staab@fsmi.ur Lukas Bach. lukas.bach@student.kit.

Übersetzungen

Homomorphismen

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

$$\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$$

- 2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?
- \rightarrow aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$ ε -Freiheit hat also die Eindeutigkeit zerstört!
- 3. Kann h aus 2 eine Codierung sein?
- → Nein, da nicht injektiv!
- 4. Warum will man ε -freie Homomorphismen?
- → Information geht sonst verloren!



Sei *h* ein Homomorphismus.

Maximilian Staab maximilian.staab@fsmi.ur Lukas Bach. lukas.bach@student.kit.

Übersetzungen

Speicher

Homomorphismen

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

$$\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$$

2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?

 \rightarrow aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$ ε -Freiheit hat also die Eindeutigkeit zerstört!

- 3. Kann h aus 2 eine Codierung sein?
- → Nein, da nicht injektiv!
- 4. Warum will man ε -freie Homomorphismen?
- → Information geht sonst verloren!
- 5. Was heißt hier Information geht verloren"?

Sei *h* ein Homomorphismus.

Maximilian Staab maximilian.staab@fsmi.ur Lukas Bach. lukas.bach@student.kit.

Übersetzungen

Speicher

Homomorphismen

Übung zu Homomorphismen

1. h(a) = 001 und h(b) = 1101. Was ist dann h(bba)?

$$\rightarrow h(bba) = h(b)h(b)h(a) = 1101 \cdot 1101 \cdot 001 = 11011101001$$

2. Sei h(a) = 01, h(b) = 11 und $h(c) = \varepsilon$. Nun sei h(w) = 011101. Was war w?

 \rightarrow aba oder cabccac, ... Allgemein: $w \in \{c\}^* \cdot \{a\} \cdot \{c\}^* \cdot \{b\} \cdot \{c\}^* \cdot \{a\} \cdot \{c\}^*$ ε -Freiheit hat also die Eindeutigkeit zerstört!

- 3. Kann h aus 2 eine Codierung sein?
- → Nein, da nicht injektiv!
- 4. Warum will man ε -freie Homomorphismen?
- → Information geht sonst verloren!
- 5. Was heißt hier Information geht verloren"?
- \rightarrow Es gibt $w_1 \neq w_2$ mit $h(w_1) = h(w_2)$

Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Information kann auch anders "verloren"gehen

Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Information kann auch anders "verloren"gehen

Maximilian Staab, maximilian staab@fsmi.uni-kar}sr \mathbf{z} h \mathbf{B} deh(a)=0, h(b)=1, h(c)=10 Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Information kann auch anders "verloren"gehen

Maximilian Staab, maximilian staab@fsmi.uni-karysrzhBdeh(a)=0, h(b)=1, h(c)=10 — Wie das? Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian.staab@fsmi.uni-kar}srzhBdeh(a)=0, h(b)=1, h(c)=10 — Wie das? Lukas Bach,

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Homomorphismen

Huffman Codierung

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian.staab@fsmi.uni-kar $\}$ srzhBdeh(a)=0, h(b)=1, h(c)=10 — Wie das?

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Huffman Codierung

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian.staab@fsmi.uni-karysrzh $B^{de}h(a)=0, h(b)=1, h(c)=10$ — Wie das? Lukas Bach,

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt

Huffman Codierun

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian.staab@fsmi.uni-karysrzh $B^{ ext{de}}h(a)=0, h(b)=1, h(c)=10$ — Wie das? Lukas Bach,

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt, dass $h(x_1)$ Präfix von

Huffman Codierung

 $h(x_2)$ ist

Information kann auch anders "verloren"gehen

Maximilian Staab

maximilian.staab@fsmi.uni-karsr \mathbf{Z} h \mathbf{B} deh(a)=0, h(b)=1, h(c)=10 — Wie das? Lukas Bach.

lukas bach@student kit edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt, dass $h(x_1)$ Präfix von $h(x_2)$ ist, dann ist h präfixfrei.

Huffman Codierung

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian.staab@fsmi.uni-karysrzh $B^{de}h(a)=0, h(b)=1, h(c)=10$ — Wie das? Lukas Bach,

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt, dass $h(x_1)$ Präfix von

Huffman Codierun

 $h(x_2)$ ist, dann ist h präfixfrei.

Speicher

Satz

Präfixfreie Codes sind injektiv.

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian. staab@fsmi.uni-kar $\mathfrak P$ srzhBdeh(a)=0, h(b)=1, h(c)=10 — Wie das? Lukas Bach,

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt, dass $h(x_1)$ Präfix von

Huffman Codierur

 $h(x_2)$ ist, dann ist h präfixfrei.

Speicher

Satz

Präfixfreie Codes sind injektiv.

Beispiele

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian.staab@fsmi.uni-kar}srzhBdeh(a)=0, h(b)=1, h(c)=10 — Wie das?

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt, dass $h(x_1)$ Präfix von $h(x_2)$ ist, dann ist h präfixfrei.

Huffman Codierung

Speicher

Satz

Präfixfreie Codes sind injektiv.

Beispiele

• h(a) = 01 und h(b) = 1101 ist präfixfrei

Information kann auch anders "verloren"gehen

Maximilian Staab,

maximilian. staab@fsmi.uni-kar $\mathfrak s$ srzhBdeh(a)=0, h(b)=1, h(c)=10 — Wie das? Lukas Bach.

lukas.bach@student.kit.edu

Präfixfreiheit

Übersetzungen

Gegeben ist ein Homomorphismus $h: A^* \to B^*$.

Homomorphismen

Wenn für keine zwei verschiedenen $x_1, x_2 \in A$ gilt, dass $h(x_1)$ Präfix von $h(x_2)$ ist, dann ist h präfixfrei.

Huffman Codierung

Speicher

Satz

Präfixfreie Codes sind injektiv.

Beispiele

- h(a) = 01 und h(b) = 1101 ist präfixfrei
- g(a) = 01 und g(b) = 011 ist nicht präfixfrei

Huffman-Codierung



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Huffman-Codierung



Maximilian Staab,

 $\substack{\text{maximilian.st} \text{aab@fsmi.uni-karlsr} \text{Komprimiert eine Zeichenkette} \\ \text{Lukas Bach,}}$

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Huffman-Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsrKte.deprimiert eine Zeichenkette Lukas Bach,

lukas.bach@student.kit.edu

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

Übersetzungen

Homomorphismen

Huffman Codierung

Huffman-Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsrKo.deprimiert eine Zeichenkette Lukas Bach,

 ${\tt lukas.bach@student.kit.edu}$

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

Übersetzungen

Homomorphismen

Vorgehensweise:

Huffman Codierung

Huffman-Codierung



Maximilian Staab.

lukas bach@student kit edu

maximilian.staab@fsmi.uni-karlsrKo.deprimiert eine Zeichenkette

Zeichen die seltener vorkommen.

Übersetzungen

Homomorphismen

Huffman Codierung

Vorgehensweise:

1. Zähle Häufigkeiten aller Zeichen der Zeichenkette

Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als

Huffman-Codierung



Maximilian Staab.

lukas bach@student kit edu

maximilian.staab@fsmi.uni-karlsrKomprimiert eine Zeichenkette

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

- Vorgehensweise:
 - 1. Zähle Häufigkeiten aller Zeichen der Zeichenkette
 - 2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander

Huffman-Codierung



Maximilian Staab.

lukas bach@student kit edu

maximilian.staab@fsmi.uni-karlsrKomprimiert eine Zeichenkette

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

- Vorgehensweise:
 - 1. Zähle Häufigkeiten aller Zeichen der Zeichenkette
 - 2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander
 - 3. Wiederhole, bis der Baum fertig ist:

Huffman-Codierung



Maximilian Staab.

lukas bach@student kit edu

maximilian.staab@fsmi.uni-karlsrKomprimiert eine Zeichenkette

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

Vorgehensweise:

- 1. Zähle Häufigkeiten aller Zeichen der Zeichenkette
- 2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander
- 3. Wiederhole, bis der Baum fertig ist:
 - Verbinde die zwei Zeichen mit niedrigsten Häufigkeiten zu neuem Knoten über diesen

Übersetzungen

Homomorphismen

Huffman Codierung

Huffman-Codierung



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsrKomprimiert eine Zeichenkette lukas bach@student kit edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

- Vorgehensweise:
 - 1. Zähle Häufigkeiten aller Zeichen der Zeichenkette
 - 2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander
 - 3. Wiederhole, bis der Baum fertig ist:
 - Verbinde die zwei Zeichen mit niedrigsten Häufigkeiten zu neuem Knoten über diesen
 - Dieser hat als Zahl die aufsummierte Häufigkeiten

Huffman-Codierung



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsrKomprimiert eine Zeichenkette lukas bach@student kit edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

- Vorgehensweise:
 - 1. Zähle Häufigkeiten aller Zeichen der Zeichenkette
 - 2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander
 - 3. Wiederhole, bis der Baum fertig ist:
 - Verbinde die zwei Zeichen mit niedrigsten Häufigkeiten zu neuem Knoten über diesen
 - Dieser hat als Zahl die aufsummierte Häufigkeiten
 - 4. Danach: Alle linken Kanten werden mit 0 kodiert, alle rechten Kanten mit 1

Huffman-Codierung



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsrKomprimiert eine Zeichenkette

lukas bach@student kit edu

 Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als Zeichen die seltener vorkommen.

Übersetzungen

Vorgehensweise:

Homomorphismen

1. Zähle Häufigkeiten aller Zeichen der Zeichenkette

Huffman Codierung

2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander

Speicher

- 3. Wiederhole, bis der Baum fertig ist:
 - Verbinde die zwei Zeichen mit niedrigsten Häufigkeiten zu neuem Knoten über diesen
 - Dieser hat als Zahl die aufsummierte Häufigkeiten
- 4. Danach: Alle linken Kanten werden mit 0 kodiert, alle rechten Kanten mit 1

Das Ergebnis ist eine Zeichenkette aus {0, 1}

Huffman-Codierung



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsrKomprimiert eine Zeichenkette

lukas bach@student kit edu

Kodiert häufiger vorkommende Zeichen zu kürzeren Codewörter als

Zeichen die seltener vorkommen.

Übersetzungen

Vorgehensweise:

Homomorphismen

1. Zähle Häufigkeiten aller Zeichen der Zeichenkette

Huffman Codierung

2. Schreibe alle vorkommenden Zeichen und ihre Häufigkeiten nebeneinander

Speicher

- 3. Wiederhole, bis der Baum fertig ist:
 - Verbinde die zwei Zeichen mit niedrigsten Häufigkeiten zu neuem Knoten über diesen
 - Dieser hat als Zahl die aufsummierte Häufigkeiten
- 4. Danach: Alle linken Kanten werden mit 0 kodiert, alle rechten Kanten mit 1

Das Ergebnis ist eine Zeichenkette aus {0, 1}, die kürzer ist als die ursprüngliche Zeichenkette in binär.

Huffman-Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Gegeben

Übersetzungen

w ∈ A*

Homomorphismen

 \mathbf{w} = afebfecaffdeddccefbeff

Huffman Codierung

Huffman-Codierung



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.ed@egeben

Übersetzungen

 $\mathbf{w} \in A^* \mathbf{w} = \text{afebfecaffdeddccefbeff}$

Homomorphismen

• Anzahl der Vorkommen aller Zeichen in w $(N_x(w))$

Huffman Codierung

Häufigkeiten:

b d Х $N_{x}(w)$

Huffman-Codierung



Maximilian Staab, Gegeben

maximilian.staab@fsmi.uni-karlsruhe.de,

lukas.bach@student.kit.edu

 $\mathbf{w} \in A^* \mathbf{w}$ = afebfecaffdeddccefbeff

• Anzahl der Vorkommen aller Zeichen in w $(N_x(w))$

Übersetzungen

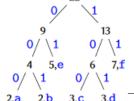
Homomorphismen

Huffman Codierung

Speicher

Zwei Phasen zur Bestimmung eines Huffman-Codes

- 1. Konstruieren eines "Baumes"
 - Blätter entsprechen den Zeichen
 - Kanten mit 0 und 1 beschriften



Häufigkeiten:

Х	а	b	С	d	е	f
$N_{x}(w)$	2	2	3	3	5	7

Huffman-Codierung



Maximilian Staab,
maximilian.staab@fsmi.uni-karg.rume.de

Lukas Bach,
lukas, bach@student.kit.edu $\mathbf{w} \in A^* \mathbf{w} = \text{afebfecaffdeddccefbeff}$

lukas.bach@student.kit.e

• Anzahl der Vorkommen aller Zeichen in w $(N_x(w))$

Übersetzungen

Zwei Phasen zur Bestimmung eines Huffman-Codes

Homomorphisme

Konstruieren eines "Baumes"

Huffman Codierung

Blätter entsprechen den Zeichen
 Kanten mit 0 und 1 beschriften

Speicher

2. Ablesen der Codes aus dem Baum (Pfadbeschriftungen)

Häufigkeiten:

Х	а	b	С	d	е	f
$N_{x}(w)$	2	2	3	3	5	7

Codewörter:

X	а	b	С	d	е	f
h(x)	000	001	100	101	01	11

Übung zu Huffman Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Übung

Übersetzungen

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

Homomorphismen

Codiere das Wort badcfehg mit Hilfe der Huffman-Codierung

Huffman Codierung

Übung zu Huffman Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Übung

Übersetzungen

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

Homomorphismen

Codiere das Wort badcfehg mit Hilfe der Huffman-Codierung

Huffman Codierung

→ Mögliche Lösung: 001 100 010 011 101 000 111 110

Übung zu Huffman Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Übung

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

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

- Codiere das Wort badcfehg mit Hilfe der Huffman-Codierung
- → Mögliche Lösung: 001 100 010 011 101 000 111 110
 - Wie lauten die Codewörter, wenn für das Wort w gilt: $N_a(w) = 1$, $N_b(w) = 2$, $N_c(w) = 2$, $N_d(w) = 8$, $N_e(w) = 16$, $N_f(w) = 16$

$$N_a(w) = 1, N_b(w) = 2, N_c(w) = 2, N_d(w) = 8, N_e(w) = 16, N_f(w) = 32, N_g(w) = 64, N_h(w) = 128$$

Übung zu Huffman Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Übung

Übersetzungen

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

Homomorphisme

Codiere das Wort badcfehg mit Hilfe der Huffman-Codierung

Huffman Codierung

→ Mögliche Lösung: 001 100 010 011 101 000 111 110

Speicher

■ Wie lauten die Codewörter, wenn für das Wort
$$w$$
 gilt: $N_a(w) = 1$, $N_b(w) = 2$, $N_c(w) = 2$, $N_d(w) = 8$, $N_e(w) = 16$, $N_f(w) = 32$, $N_g(w) = 64$, $N_h(w) = 128$

Mögliche Lösung:

Х	а	b	С	d	е	f	g	h
h(x)	0000000	0000001	000001	00001	0001	001	01	1

Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Wie lang w\u00e4re das zweite Wort (abbcccc d⁸...g⁶⁴h¹²⁸) mit dem ersten Code codiert?

Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Wie lang w\u00e4re das zweite Wort (abbcccc d⁸...g⁶⁴h¹²⁸) mit dem ersten Code codiert?

Homomorphismen

ightarrow 741 Symbole. Also dreimal so lang wie das Original.

Huffman Codierung

Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de. Lukas Bach.

lukas bach@student kit edu

Übersetzungen

■ Wie lang wäre das zweite Wort (abbcccc d⁸...g⁶⁴h¹²⁸) mit dem ersten Code codiert?

Homomorphismen

→ 741 Symbole. Also dreimal so lang wie das Original.

Huffman Codierung

Wie lang wäre das zweite Wort mit dem zweiten Code codiert?

Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Wie lang wäre das zweite Wort (abbcccc d⁸...g⁶⁴h¹²⁸) mit dem ersten Code codiert?

Homomorphismen

ightarrow 741 Symbole. Also dreimal so lang wie das Original.

Huffman Codierung

Wie lang wäre das zweite Wort mit dem zweiten Code codiert?

Speicher

ightarrow 501 Symbole. Also nur zweimal so lang wie das Original.

Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Wie lang wäre das zweite Wort (abbcccc d⁸...g⁶⁴h¹²⁸) mit dem ersten Code codiert?

Homomorphismen

ightarrow 741 Symbole. Also dreimal so lang wie das Original.

Huffman Codierung

Wie lang wäre das zweite Wort mit dem zweiten Code codiert?

Speicher

ightarrow 501 Symbole. Also nur zweimal so lang wie das Original.

Was fällt euch auf?

Wahr oder falsch?



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas bach@student.kit.edu

Sei $h: A^* \to \mathbb{Z}_2$ eine Huffman-Codierung

Übersetzungen

• h ist ein ε -freier Homomorphismus

Homomorphismen

Huffman Codierung

Wahr oder falsch?



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Sei $h:A^* \to \mathbb{Z}_2$ eine Huffman-Codierung

Übersetzungen

h ist ein ε-freier Homomorphismus Wahr!

Homomorphismen

Huffman Codierung

Wahr oder falsch?



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas bach@student kit edu

Sei $h: A^* \to \mathbb{Z}_2$ eine Huffman-Codierung

• h ist ein ε -freier Homomorphismus **Wahr!**

Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren

Huffman Codierung

Speicher

Übersetzungen

Homomorphismen

Wahr oder falsch?



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de. Lukas Bach.

lukas bach@student kit edu

Sei $h: A^* \to \mathbb{Z}_2$ eine Huffman-Codierung

Übersetzungen

Homomorphismen

Huffman Codierung

- h ist ein ε -freier Homomorphismus **Wahr!**
- Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren Falsch!

Wahr oder falsch?



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

- h ist ein ε -freier Homomorphismus **Wahr!**
 - Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren Falsch!
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind.

Wahr oder falsch?



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

- h ist ein ε -freier Homomorphismus **Wahr!**
- Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren Falsch!
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind. Falsch!

Wahr oder falsch?



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

- h ist ein ε -freier Homomorphismus **Wahr!**
 - Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren Falsch!
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind. Falsch!
- h ist präfixfrei

Wahr oder falsch?



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

- h ist ein ε -freier Homomorphismus **Wahr!**
- Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren Falsch!
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind. Falsch!
- h ist präfixfrei Wahr!

Wahr oder falsch?



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

- h ist ein ε -freier Homomorphismus **Wahr!**
- Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren Falsch!
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind. Falsch!
- h ist präfixfrei Wahr!
- Es kann noch kürzere Codierungen geben

Wahr oder falsch?



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

- h ist ein ε-freier Homomorphismus Wahr!
- Häufigere Symbole werden mit langen Worten codiert, seltene mit kürzeren Falsch!
- Die Kompression ist am stärksten, wenn die Häufigkeiten aller Zeichen ungefähr gleich sind. Falsch!
- h ist präfixfrei Wahr!
- Es kann noch kürzere Codierungen geben Falsch!

Huffman-Codierung



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Eigenschaften

Sei A ein Alphabet und $w \in A$. Dann gilt für die Huffman-Codierung h:

- $h: A^* \to \mathbb{Z}_2$
- h ist ε -freier Homomorphismus
- h ist präfixfreier Homomorphismus
- Häufigere Symbole werden mit kurzen Worten codiert, seltene mit längeren
- Produziert kürzestmögliche Codierungen

Block-Codierung mit Huffman



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Block-Codierung mit Huffman



Maximilian Staab,

maximilian.staab@fsmi.uni-kaulsrWirdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach,

lukas.bach@student.kit.edu fester Länge b>1

Übersetzungen

Homomorphismen

Huffman Codierung

Block-Codierung mit Huffman



Maximilian Staab,

maximilian.staab@fsmi.uni-kamsstWfrdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach,

lukas.bach@student.kit.edu fester Länge b>1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Huffman Codierung

Beispiel an der Tafel: Codierung von

 $aab \cdot deg \cdot deg \cdot aab \cdot ole \cdot aab \cdot deg \cdot aab.$

Block-Codierung mit Huffman



Maximilian Staab,

maximilian.staab@fsmi.uni-kamsstWfrdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach,

lukas.bach@student.kit.edu fester Länge b>1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Huffman Codierung

Beispiel an der Tafel: Codierung von

 $aab \cdot deg \cdot deg \cdot aab \cdot ole \cdot aab \cdot deg \cdot aab.$

Block-Codierung mit Huffman



Maximilian Staab,

maximilian.staab@fsmi.uni-kamsstWfrdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach,

lukas.bach@student.kit.edu fester Länge b>1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Huffman Codierung

Beispiel an der Tafel: Codierung von

 $aab \cdot deg \cdot deg \cdot aab \cdot ole \cdot aab \cdot deg \cdot aab.$

Block-Codierung mit Huffman



Maximilian Staab,

maximilian.staab@fsmi.uni-kamestWirdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach.

Ikas bach@student.kit.edu

fester Länge h > 1

.bach@student.kit.edu fester Länge b>1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Huffman Codierung

Beispiel an der Tafel: Codierung von

aab · deg · deg · aab · ole · aab · deg · aab.

Speicher

• Alphabet $A = \{a, b, c, d\}$

Block-Codierung mit Huffman



Maximilian Staab,

maximilian.staab@fsmi.uni-kadstudern.blöcke von Lukas Bach,
lukas.bach@student.kit.edu fester Länge b > 1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Huffman Codierung

Beispiel an der Tafel: Codierung von

aab · deg · deg · aab · ole · aab · deg · aab.

- Alphabet $A = \{a, b, c, d\}$
- Text über A, der nur aus Teilwörtern der Länge 10 zusammengesetzt ist, in denen jeweils immer nur ein Symbol vorkommt

Block-Codierung mit Huffman



Maximilian Staab.

maximilian.staab@fsmi.uni-kawiszWirdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach. lukas.bach@student.kit.edu fester Länge b > 1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Beispiel an der Tafel: Codierung von Huffman Codierung aab · deg · deg · aab · ole · aab · deg · aab.

- Alphabet $A = \{a, b, c, d\}$
- Text über A, der nur aus Teilwörtern der Länge 10 zusammengesetzt ist, in denen jeweils immer nur ein Symbol vorkommt
- Angenommen a¹⁰, ..., d¹⁰ kommen alle gleich häufig vor. Wie lang ist dann die Huffman-Codierung?

Block-Codierung mit Huffman



Maximilian Staab.

maximilian.staab@fsmi.uni-kawiszWirdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach. lukas.bach@student.kit.edu fester Länge b > 1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Beispiel an der Tafel: Codierung von Huffman Codierung aab · deg · deg · aab · ole · aab · deg · aab.

- Alphabet $A = \{a, b, c, d\}$
- Text über A, der nur aus Teilwörtern der Länge 10 zusammengesetzt ist, in denen jeweils immer nur ein Symbol vorkommt
- Angenommen a¹⁰, ..., d¹⁰ kommen alle gleich häufig vor. Wie lang ist dann die Huffman-Codierung?

Block-Codierung mit Huffman



Maximilian Staab

maximilian.staab@fsmi.uni-kawiszWirdbetrachten nicht mehr einzelne Symbole, sondern Blöcke von Lukas Bach.

lukas.bach@student.kit.edu fester Länge b > 1

Übersetzungen

Blätter des Huffman-Baums sind jetzt Wörter der Länge b

Homomorphismen

Beispiel an der Tafel: Codierung von Huffman Codierung aab · deg · deg · aab · ole · aab · deg · aab.

- Alphabet $A = \{a, b, c, d\}$
- Text über A, der nur aus Teilwörtern der Länge 10 zusammengesetzt ist, in denen jeweils immer nur ein Symbol vorkommt
- Angenommen a¹⁰, ..., d¹⁰ kommen alle gleich häufig vor. Wie lang ist dann die Huffman-Codierung?
- → Ein Fünftel, weil jeder Zehnerblock durch zwei Bits codiert wird

Speicher



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen Ein Bit ist Zeichen aus $A = \{0, 1\}$

Huffman Codierung

Speicher



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

• Ein **Bit** ist Zeichen aus $A = \{0, 1\}$

Huffman Codierung

■ Ein **Byte** ist ein Wort aus acht Bits

Speicher



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

• Ein **Bit** ist Zeichen aus $A = \{0, 1\}$

Huffman Codierung

■ Ein **Byte** ist ein Wort aus acht Bits

Abkürzungen

Speicher



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

• Ein **Bit** ist Zeichen aus $A = \{0, 1\}$

Huffman Codierung

Ein Byte ist ein Wort aus acht Bits

Speicher

Abkürzungen

Für Bit: bit

Speicher



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Ein **Bit** ist Zeichen aus $A = \{0, 1\}$

Huffman Codierung

■ Ein Byte ist ein Wort aus acht Bits

Speicher

Abkürzungen

Für Bit: bit

Für Byte: B

Präfixe



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Dezimal

	Dezimai						
	10^{-3} 10^{-6}		10^{-9} 10^{-12}		10 ⁻¹⁵	10^-18	
	1000^{-1}	1000^{-2}	1000^{-3}	1000^{-4}	1000^{-5}	1000^{-6}	
	milli	mikro	nano	pico	femto	atto	
	m	μ	n p		f	a	
	10 ³	10 ³ 10 ⁶		10 ¹²	10 ¹⁵	10 ¹⁸	
	1000^{1}	1000^{2}	1000^{3}	1000^{4}	1000^{5}	1000^{6}	
	kilo	kilo mega		tera	peta	exa	
	k	M	G	T	P	E	
-							

Präfixe



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Dezimal

B O Z I I I I I							
10-3	10^{-6}	10^{-9} 10^{-12}		10^{-15}	10^{-18}		
1000^{-1}	1000^{-2}	1000^{-3}	1000^{-4}	1000^{-5}	1000^{-6}		
milli	mikro	ro nano pico n p		femto atto			
m	μ			f	a		
10 ³	10 ⁶	109	10 ¹²	10 ¹⁵	10 ¹⁸		
1000^{1}	1000^{2}	1000^{3}	1000^{4}	1000^{5}	1000^{6}		
kilo	mega	giga	tera	peta	exa		
k	M	G	T	P	E		

Binär

2 ¹⁰	2^{20}	2^{30} 2^{40}		2^{50}	2^{60}		
1024^{1}	1024^{2}	1024^{3}	1024^{4}	1024^{5}	1024^{6}		
kibi	mebi	gibi	tebi	pebi	exbi		
Ki	Mi	Gi	Ti	Pi	Ei		

Gesamtzustand eines Speichers



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Gesamtzustand eines Speichers



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edZu jedem Zeitpunkt ist

Übersetzungen

Homomorphismen

Huffman Codierung

Gesamtzustand eines Speichers



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edZu jedem Zeitpunkt ist

■ für jede Adresse festgelegt, welcher Wert dort ist

Übersetzungen

Homomorphismen

Huffman Codierung

Gesamtzustand eines Speichers



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach, lukas.bach@student.kit.edZu jedem Zeitpunkt ist

Übersetzungen

für jede Adresse festgelegt, welcher Wert dort ist

beides meist Bitfolgen

Homomorphismen

Huffman Codierung

Gesamtzustand eines Speichers



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edZu jedem Zeitpunkt ist

Übersetzungen

 für jede Adresse festgelegt, welcher Wert dort ist beides meist Bitfolgen

Homomorphismen

Vorstellung: Tabelle mit zwei Spalten

Huffman Codierung

Adresse	Wert			
Adresse 1	Wert 1			
Adresse 2	Wert 2			
Adresse 3	Wert 3			
Adresse n	Wert n			

Zustand eines Speichers – formal



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Zustand eines Speichers – formal



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Ü	h		re		+>		n	α	
U	U	U	10	C	LZ	u	11	У	C

Definition des Speicherzustandes

Homomorphismen

Sei Adr die Menge aller Adressen und Val die Menge aller Werte.

Huffman Codierung

Dann ist

Speicher

 $m: Adr \rightarrow Val$

der aktuelle Zustand des Speichers. Dabei ist m(a) der aktuelle Wert an der Adresse a.

Lesen und Speichern



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Lesen und Speichern



Maximilian Staab, maximilian.staab@fsmi.ur

Mem

Lukas Bach, lukas.bach@student.kit.

Menge aller möglichen Speicherzustände, also Menge aller Abbildungen von *Adr* nach *Val*

Übersetzungen

 $Mem := Val^{Adr}$

Homomorphismen

Huffman Codierung

Lesen und Speichern



Maximilian Staab, maximilian.staab@fsmi.ur

Mem

Lukas Bach, lukas.bach@student.kit.

Menge aller möglichen Speicherzustände, also Menge aller Abbildungen von *Adr* nach *Val*

Übersetzungen

 $Mem := Val^{Adr}$

Homomorphismen

Anmerkung:

Huffman Codierun

Lesen und Speichern



Maximilian Staab, maximilian.staab@fsmi.u

Mem

Lukas Bach, lukas.bach@student.kit.

Menge aller möglichen Speicherzustände, also Menge aller Abbildungen von *Adr* nach *Val*

Übersetzungen

 $Mem := Val^{Adr}$

Homomorphismen

Anmerkung: Für zwei Mengen A, B gilt

Huffman Codierun

Lesen und Speichern



Maximilian Staab, maximilian.staab@fsmi.u

Mem

Lukas Bach, lukas.bach@student.kit.

Menge aller möglichen Speicherzustände, also Menge aller Abbildungen von *Adr* nach *Val*

Übersetzungen

 $Mem := Val^{Adr}$

Homomorphismen

Anmerkung: Für zwei Mengen A, B gilt: $A^B := \{f : B \to A\}$.

Huffman Codierun

Lesen und Speichern



Maximilian Staab, maximilian.staab@fsmi.u Lukas Bach.

lukas.bach@student.kit.

Mem

Menge aller möglichen Speicherzustände, also Menge aller Abbildungen von *Adr* nach *Val*

Übersetzungen

Homomorphismen

Huffman Codierung

 $Mem := Val^{Adr}$

Anmerkung: Für zwei Mengen A, B gilt: $A^B := \{f : B \to A\}$.

memread

Speicher

 $memread: Mem \times Adr \rightarrow Val \; mit \; (m,a) \mapsto m(a)$

Lesen und Speichern



Maximilian Staab, maximilian.staab@fsmi.u Lukas Bach.

Mem

Menge aller möglichen Speicherzustände, also Menge aller Abbildungen von Adr nach Val

Übersetzungen

Homomorphismen

lukas.bach@student.kit.

Huffman Codierung

 $Mem := Val^{Adr}$

Anmerkung: Für zwei Mengen A, B gilt: $A^B := \{f : B \to A\}$.

memread

 $memread: Mem \times Adr \rightarrow Val \ mit \ (m, a) \mapsto m(a)$

Speicher

memwrite

memwrite : $\textit{Mem} \times \textit{Adr} \times \textit{Val} \rightarrow \textit{Mem} \ \text{mit} \ (\textit{m},\textit{a},\textit{v}) \mapsto \textit{m}'$

Für m' wird folgendes gefordert:

$$m(a') := egin{cases} v & ext{falls } a' = a \ m(a') & ext{falls } a'
eq a \end{cases}$$

Eigenschaften von memread und memwrite



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Eigenschaften von memread und memwrite



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Eigenschaften ("Invarianten")

lacktriangledown memread(memwrite(m, a, v), a) = v

Eigenschaften von memread und memwrite



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierun

Speicher

Eigenschaften ("Invarianten")

■ memread(memwrite(m, a, v), a) = v (Also: An a einen Wert v zu schreiben und danach bei a zu lesen gibt den Wert v zurück

Eigenschaften von memread und memwrite



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphismen

Huffman Codierung

Speicher

Eigenschaften ("Invarianten")

■ memread(memwrite(m, a, v), a) = v (Also: An a einen Wert v zu schreiben und danach bei a zu lesen gibt den Wert v zurück ⇒ Konsistente Datenhaltung)

Eigenschaften von memread und memwrite



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphisme

Huffman Codierun

Speicher

- memread(memwrite(m, a, v), a) = v (Also: An a einen Wert v zu schreiben und danach bei a zu lesen gibt den Wert v zurück ⇒ Konsistente Datenhaltung)
- memread(memwrite(m, a', v'), a) = memread(m, a)

Eigenschaften von memread und memwrite



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphisme

Huffman Codierun

Speicher

- memread(memwrite(m, a, v), a) = v (Also: An a einen Wert v zu schreiben und danach bei a zu lesen gibt den Wert v zurück ⇒ Konsistente Datenhaltung)
- memread(memwrite(m, a', v'), a) = memread(m, a) (Also: Auslesen einer Speicherstelle ist unabhängig davon, was vorher an eine andere Adresse geschrieben wurde

Eigenschaften von memread und memwrite



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Übersetzungen

Homomorphisme

Huffman Codierun

Speicher

- memread(memwrite(m, a, v), a) = v (Also: An a einen Wert v zu schreiben und danach bei a zu lesen gibt den Wert v zurück ⇒ Konsistente Datenhaltung)
- memread(memwrite(m, a', v'), a) = memread(m, a) (Also: Auslesen einer Speicherstelle ist unabhängig davon, was vorher an eine andere Adresse geschrieben wurde \Rightarrow Unabhängige Datenhaltung)

Maximilian Staab. maximilian.staab@fsmi.uni-karlgaben

Lukas Bach, Aktueller Speicherzustand:

Adresse Wert Übersetzungen 01110 00000 Homomorphismen 00001 00100 00010 00111 Huffman Codierung 00011 00000 Speicher

Was ist?

memread(memwrite(m, memread(m, 00011), 01010), 00000)

Maximilian Staab. maximilian.staab@fsmi.uni-karlgaben

Lukas Bach, Aktueller Speicherzustand:

Übersetzungen Homomorphismen Huffman Codierung Adresse Wert 01110 00000 00001 00100 00010 00111 00011 00000

Speicher

Was ist?

- memread(memwrite(m, memread(m, 00011), 01010), 00000)
- \rightarrow 01010

Maximilian Staab,

maximilian.staab@fsmi.uni Lukas Bach,

lukas.bach@student.kit.ed

Übersetzungen

Homomorphismen

Huffman Codierung

