Wann sind Codes eindeutig entschlüsselbar?

Definition Suffix

Sei C ein Code. Ein Folge $s \in \{0,1\}^*$ heißt Suffix in C falls

1. $\exists c_i, c_j \in C : c_i = c_j s$ oder
2. $\exists c \in C$ und einen Suffix s' in C: $s' = cs$ oder
3. $\exists c \in C$ und einen Suffix s' in C: $c = s's$.

- Bedingung 1: Codewort c_j lässt sich zu Codewort c_i erweitern.
- Bedingung 2: Codewort c lässt sich zu Suffix s' erweitern.
- Bedingung 3: Suffix s' lässt sich zu Codewort c erweitern.

Effiziente Berechnung von Suffixen

Algorithmus Berechnung Suffix

1. Setze $S := \emptyset, T := \emptyset$.
2. Für alle $c_i, c_j \in C \times C$: Falls es ein $s \in \{0,1\}^*$ gibt mit $c_i = c_j s$, füge s in S und T ein.
3. Solange $T \neq \emptyset$
   1. Entferne ein beliebiges s' aus T.
   2. Für alle $c \in C$: Falls es ein $s \in \{0,1\}^* \setminus S$ gibt mit $c = s's$ oder $s' = cs$, füge s zu S und T hinzu.

Laufzeit: $C = \{c_1, \ldots, c_n\}$

- Schritt 2: $\mathcal{O}(n^2)$ Codewortpaare
- Suffixlänge ist durch $\max_i\{|c_i|\}$ beschränkt.
- Es kann höchstens $n \cdot \max_i\{|c_i|\}$ Suffixe geben.
- Schritt 3: $\mathcal{O}(n^2 \cdot \max_i\{|c_i|\})$
- **Polynomiell in der Eingabelänge**: n, $max_i\{|c_i|\}$

Beispiele Suffixberechnung

- Code $C_2 = \{0, 1, 00\}$
  - Suffix $s_1 = 0$, denn $c_3 = c_1 0$.
- Code $C_3 = \{0, 01, 011\}$
  - Suffix $s_1 = 1$, denn $c_2 = c_1 1$.
  - Suffix $s_2 = 11$, denn $c_3 = c_1 11$.
- Code $C_4 = \{0, 10, 110\}$
  - Keine Suffixe, da Präfixcode.
- Code $C_5 = \{1, 110, 101\}$
  - Suffix $s_1 = 10$, denn $c_2 = c_1 10$.
  - Suffix $s_2 = 01$, denn $c_3 = c_1 01$.
  - Suffix $s_3 = 0$, denn $s_3 = c_1 0$.
  - Suffix $s_4 = 1$, denn $c_3 = s_1 1$.

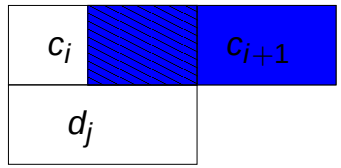
Kriterium für eindeutig entschlüsselbar

Eindeutig entschlüsselbar

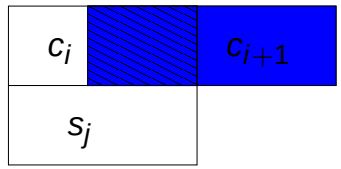
C ist ein eindeutig entschlüsselbarer Code $\Leftrightarrow$ Kein Suffix ist Codewort in C.

z.z.: C nicht eindeutig entschlüsselbar $\Rightarrow$ Suffix ist Codewort

- Zwei gleiche Folgen $c_1 \ldots c_n$ und $d_1 \ldots d_m$ von verschiedenen Codeworten
- Fall 1: Codewort c_i lässt sich zu d_j erweitern

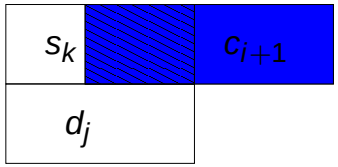


- Fall 2: Codewort c_i lässt sich zu Suffix s_j erweitern



Suffix ist Codewort

- Fall 3: Suffix s_k lässt sich zu Codewort d_j erweitern



- Nach jedem Schritt beginnt der konstruierte Suffix mit einem Codewortpräfix.
- Der zuletzt konstruierte Suffix ist identisch mit dem letzten Codewort von beiden Sequenzen.

Rückrichtung

z.z.: Suffix s ist ein Codewort $\Rightarrow$ C ist nicht eindeutig entschlüsselbar

- Suffix s ist aus Anwendungen der drei Regeln entstanden.
- Berechne die Kette zurück, aus der s entstanden ist.
  - Setze String $c^* \leftarrow s$. Iteriere:
  - 1. Fall $c_i = c_j s$: $c^* \leftarrow c_j c^*$, terminiere.
  - 2. Fall $s' = cs$: $c^* \leftarrow cc^*$, $s \leftarrow s'$.
  - 3. Fall $c = s's$: $c^* \leftarrow s'c^*$, $s \leftarrow s'$.
- Kette muss mit 1. Fall $c_i = c_j s'$ terminieren.
- Zwei verschiedene Entschlüsselungen:
  Eine beginnt mit c_i, die andere mit c_j.
- Beide sind gültig, da der letzte Suffix ein Codewort ist.

Beispiel: Für $C = 1, 110, 101$ erhalten wir für den Suffix 1 den String $c^* = 1101$ mit gültigen Dekodierungen $1|101$ und $110|1$.

Sätze von Kraft und McMillan

Satz von Kraft

Ein Präfixcode C für das Alphabet $A = \{a_1, \ldots, a_n\}$ mit Kodierungslängen $|C(a_j)| = \ell_j$ existiert gdw

$$\sum_{j=1}^{n} 2^{-\ell_j} \leq 1.$$

Satz von McMillan

Ein eindeutig entschlüsselbarer Code C für das Alphabet $A = \{a_1, \ldots, a_n\}$ mit Kodierungslängen $|C(a_j)| = \ell_j$ existiert gdw

$$\sum_{j=1}^{n} 2^{-\ell_j} \leq 1.$$

Präfixcodes genügen

Korollar

Ein Präfixcode C existiert gdw es einen eindeutig entschlüsselbaren Code C mit denselben Kodierungslängen gibt.

- Wir zeigen den Ringschluss für:
  $\sum_{j=1}^{n} 2^{-\ell_j} \leq 1 \Rightarrow$ Präfix $\Rightarrow$ Eindeutig entschlüsselbar

  (Präfix $\Rightarrow$ Eindeutig entschlüsselbar: letzte Vorlesung)
- Gegeben sind Kodierungslängen ℓ_j. Gesucht ist ein Präfixcode mit $\ell_j = |C(a_j)|$.
- Definiere $\ell := \max\{\ell_1, \ldots, \ell_n\}$, $n_i :=$ Anzahl ℓ_j mit $\ell_j = i$.

$$\sum_{j=1}^{n} 2^{-\ell_j} = \sum_{j=1}^{\ell} n_j 2^{-j} \leq 1.$$

Beweis: $\sum_{j=1}^{n} 2^{-\ell_j} \leq 1 \Rightarrow$ Präfix

Induktion über l:

- **IA** $\ell = 1$**:** $n_1 \leq 2$
- Können Präfixcode $C \subseteq \{0, 1\}$ für max. 2 Codeworte konstruieren.

- **IS** $\ell - 1 \rightarrow \ell$**:** $n_\ell \leq 2^\ell - 2^{\ell-1}n_1 - 2^{\ell-2}n_2 - \ldots 2n_{\ell-1}$
- **IV:** Präfixcode mit n_i Worten der Länge i, $i = 1, \ldots, \ell - 1$.
- Anzahl der Worte der Länge ℓ: 2^ℓ
- Anzahl der Worte der Länge ℓ mit Präfixen $n_1, \ldots, n_{\ell-1}$: $2^{\ell-1}n_1 + \cdots + 2n_{\ell-1}$.
- Können die n_ℓ Worte mit den verbleibenden $2^\ell - (2^{\ell-1}n_1 + \cdots + 2n_{\ell-1})$ Worten der Länge ℓ als Präfixcode kodieren.

Eindeutig entschlüsselbar $\Rightarrow \sum_{j=1}^{n} 2^{-\ell_j} \leq 1$

- Sei C eindeutig entschlüsselbar mit $C(a_j) = \ell_j$, $\ell = max_j\{\ell_j\}$.
- Wählen $r \in \mathbb{N}$ beliebig. Betrachten

$$\left(\sum_{j=1}^{n} 2^{-\ell_j}\right)^r = \sum_{i=1}^{r\ell} n_i 2^{-i}$$

- Analog zum Beweis zuvor: n_i = Anzahl Strings aus $\{0,1\}^i$, die sich als Folge von r Codeworten schreiben lässt.
- C eindeutig entschlüsselbar: Jeder String aus $\{0,1\}^i$ lässt sich als höchstens eine Folge von Codeworten schreiben, d.h. $n_i \leq 2^i$.
- Damit gilt $\sum_{i=1}^{r\ell} n_i 2^{-i} \leq r\ell \quad \Rightarrow \quad \sum_{j=1}^{n} 2^{-\ell_j} \leq (r\ell)^{\frac{1}{r}}$
- Für $r \to \infty$ folgt $\sum_{j=1}^{n} 2^{-\ell_j} \leq 1$.

Huffman Kodierung

Szenario: Quelle Q mit Symbole $\{a_1, \ldots, a_n\}$

- a_i sortiert nach absteigenden Quellws. $p_1 \geq p_2 \geq \cdots \geq p_n$.

Algorithmus Huffman-Kodierung

Eingabe: Symbole a_i mit absteigend sortierten p_i, $i = 1, \ldots, n$.

1. IF (n=2), Ausgabe $C(a_1) = 0$, $C(a_2) = 1$.
2. ELSE
   1. Bestimme $k \in \mathbb{Z}_{n-1}$ mit $p_k \geq p_{n-1} + p_n \geq p_{k+1}$.
   2. $(p_1, \ldots, p_k, p_{k+1}, p_{k+2}, \ldots, p_{n-1}) \leftarrow$
      $(p_1, \ldots, p_k, p_{n-1} + p_n, p_{k+1}, \ldots, p_{n-2})$
   3. $(C(a_1), \ldots, C(a_{k-1}), C(a_{k+1}), \ldots, C(a_{n-2}), C(a_k)0, C(a_k)1) \leftarrow$
      Huffmann-Kodierung$(a_1, \ldots, a_{n-1}, p_1, \ldots, p_{n-1})$

Ausgabe: kompakter Präfixcode für Q

Laufzeit: $\mathcal{O}(n^2)$ ($\mathcal{O}(n \log n)$ mit Hilfe von Heap-Datenstruktur)

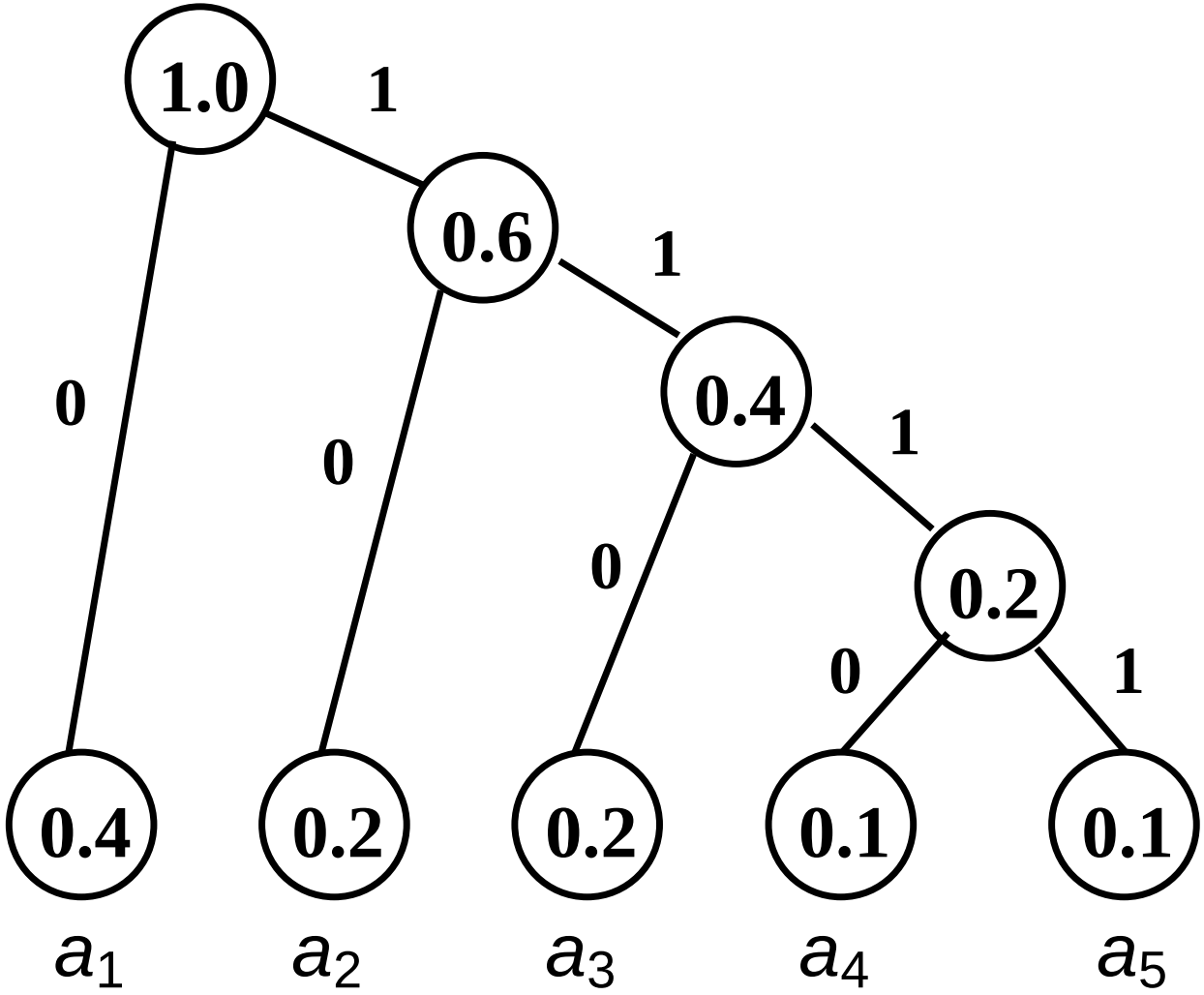
Beispiel Huffman-Kodierung

Beispiel: $p_1 = 0.4, p_2 = p_3 = 0.2, p_4 = p_5 = 0.1$

a_i	p_i	$C(a_i)$	p_i	$C(a_i)$	p_i	$C(a_i)$	p_i	$C(a_i)$
a_1	0.4	00	0.4	00	**0.4**	1	**0.6**	0
a_2	0.2	01	0.2	01	0.4	00	0.4	1
a_3	0.2	11	**0.2**	10	0.2	01		
a_4	0.1	100	0.2	11				
a_5	0.1	101						

- **Fett** gedruckt: Stelle k Man beachte: k ist nicht eindeutig, d.h. C ist nicht eindeutig.
- $E(C) = (0.4 + 0.2 + 0.2) * 2 + 2 * 0.1 * 3 = 2.2$
- Huffman-Tabelle: Spalten 1 und 3. Mittels Huffman-Tabelle kann jeder String $m \in A^*$ in Zeit $\mathcal{O}(|C(m)|)$ kodiert werden.

Wahl eines anderen k



$$E(C') = 0.4 * 1 + 0.2 * (2 + 3) + 0.1 * 2 * 4 = 2.2$$

Eigenschaften kompakter Codes

Sei $\ell_i := |C(a_i)|$.

Lemma: Eigenschaften kompakter Codes

Sei C ein kompakter Code, oBdA ist C ein Präfixcode.

1. Falls $p_i > p_j$, dann ist $\ell_i \leq \ell_j$
2. Es gibt mindestens zwei Codeworte in C mit maximaler Länge.
3. Unter den Worten mit maximaler Länge existieren zwei Worte, die sich nur in der letzten Stelle unterscheiden.

Beweis der Eigenschaften

Beweis:

1. Sei $\ell_i > \ell_j$. Dann gilt

$$\begin{aligned} & p_i\ell_i + p_j\ell_j = p_i(\ell_i - \ell_j + \ell_j) + p_j(\ell_j - \ell_i + \ell_i) \\ = \quad & p_i\ell_j + p_j\ell_i + (\ell_i - \ell_j)(p_i - p_j) > p_i\ell_j + p_j\ell_i \end{aligned}$$

   D.h. vertauschen der Kodierungen von a_i und a_j verkürzt den Code.

2. Sei $c = c_1 \dots c_n \in C$ das einzige Codewort mit maximaler Länge. Streichen von c_n führt zu einem Präfixcode mit kürzerer erwarteter Codewortlänge.
3. Annahme: Alle Paar von Codeworten maximaler Länge unterscheiden sich nicht nur in der letzten Komponente.
   - Entferne die letzte Komponente eines beliebigen Codewortes maximaler Länge.
   - Wir erhalten einen Präfixcode mit kürzerer Länge.

Optimalität der Huffman-Kodierung

Satz

Die Huffman-Kodierung liefert einen kompakten Code.

Beweis per Induktion über n.

- **IA:** $n = 2$: Für $\{a_1, a_2\}$ ist die Codierung $\{0, 1\}$ kompakt.
- **IS:** $n - 1 \rightarrow n$: Sei C' kompakt für $\{a_1, \ldots, a_n\}$.
  - Lemma,2: C' enthält zwei Codeworte maximaler Länge.
  - Lemma,3: Unter den Codeworten maximaler Länge gibt es zwei Codeworte $c0, c1 \in C'$ mit $c \in \{0, 1\}^*$, die sich nur in der letzten Stelle unterscheiden.
  - Lemma,1: Die beiden Symbole a_{n-1}, a_n mit kleinster Quellws besitzen maximale Codewortlänge. Vertausche die Kodierungen dieser Symbole mit $c0, c1$.
  - a_{n-1} oder a_n tauchen mit Ws $p_{n-1} + p_n$ auf.
  - **IA:** Huffman-Kodierung liefert kompakten Präfixcode C für $a_1, \ldots, a_{n-2}, a'$ mit Quellws $p_1, \ldots, p_{n-2}, p_{n-1} + p_n$
  - $C(a_1), \ldots, C(a_{n-2}), C(a')0 = c0, C(a')1 = c1$ ist Präfixcode mit erwarteter Codewortlänge $E(C')$, d.h. die Huffman-Kodierung liefert einen kompakten Präfixcode.