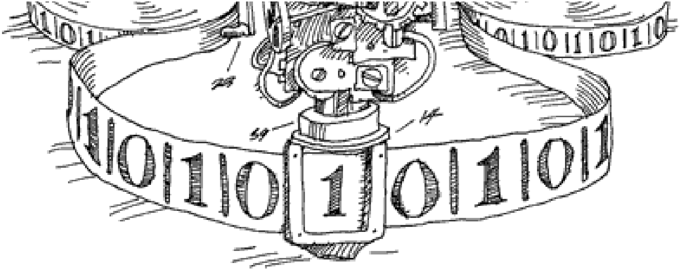
Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Tutorium 6

Institut für Kryptographie und Sicherheit



Best of Übungsblatt 3

- Abgaben: 14 von 21 (66,6%)
- Mindestens 50% Punkte: 16 von 16 (100%)
  - Auf diesem Blatt benötigt man min. 6 Punkte für 50%
- Durchschnittliche erreichte Punktzahl: 8.00
  - (nach Aufrunden auf halbe Punkte)

Aufgabenverteilung

Aufgabe 1: Gesamt: 43.25P (durchscnittlich 3.08)

Aufgabe 2: Gesamt: 11.5P (durchschnittlich 0.82)

- „Bonuspunkte“

Aufgabe 3: Gesamt: 37.25P (durchschnittlich 2.66)

Aufgabe 4: Gesamt: 19P (durchschnittlich 1.36)

Grammatiken

Eine kontextsensitive Grammatik $G = (T, V, S, P)$ (Chomsky Typ 1)

- T = Menge der Terminale (a.k.a. Alphabet der Sprache)
- V = Menge der Nichtterminale (zu T disjunkt)
- $S \in V$ = Startsymbol
- P = Menge der Produktionen mit Form $v \to w$
  - $v \in V^+$
  - $w \in ((V \setminus \{S\}) \bigcup T)^+$
  - $|v| \leq |w|$ oder $S \to \epsilon$

Eine Grammatik $G = (T, V, S, P)$ (Chomsky Typ 0)

- T = Menge der Terminale (a.k.a. Alphabet der Sprache)
- V = Menge der Nichtterminale (zu T disjunkt)
- $S \in V$ = Startsymbol
- P = Menge der Produktionen mit Form $v \to w$
  - $v, w \in (V \bigcup T)^*$

Chomsky-Hierarchie

- **Chomsky Typ 3**
  - Reguläre Sprachen (z.B. rechtslineare Sprachen)
  - Reguläre Ausdrücke
  - Endliche Automaten
- **Chomsky Typ 2**
  - $Ch3 \subseteq Ch2$
  - Kontextfreie Sprachen
  - Nichtdeterministische Kellerautomaten
  - Programmiersprachen sind in der Regel $Ch2$
- **Chomsky Typ 1**
  - $Ch2 \subseteq Ch1$
  - Kontextsensitiven Sprachen
  - Nichtdeterministische, linear platzbeschränkte Turingmaschine
- **Chomsky Typ 0**
  - $Ch1 \subseteq Ch0$
  - Grammatiken mit beliebiger Produktionsmenge
  - Semi-entscheidbare Sprachen (durch Turingmaschine)

Gödelnummer

- Jede Turingmaschine lässt sich eindeutig als natürliche Zahl darstellen.
  Diese Zahl ist ihre *Gödelnummer*.
  - Das bedeutet auch, dass die Menge aller Turingmaschinen abzählbar ist!
- Eine Möglichkeit eine Turingmaschine binär zu kodieren wäre folgende:
  Alle n Zustandsübergänge $\delta(q_i, x_j) = (q_k, x_l, d_m)$,
  mit $d_1 = L$, $d_2 = N$, $d_3 = R$
  kodieren in
  $111 code_1 11 code_2 11 ... 11 code_{n-1} 11 code_n 111$
  Wobei $code_r$ so kodiert ist:
  $0^i 1 0^j 1 0^k 1 0^l 1 0^m$
- Eine solche eindeutige Kodierung nennen wir *Gödelisierung*.

Universelle Turingmaschine

Karlsruher Institut für Technologie

- Eingabe:
  1. Kodierung einer TM
  2. Eingabe für die zu simulierende TM
- Simulation der übergebenen TM
- Ausgabe: Ausgabe der simulierten TM.

Definitionen zur Entscheidbarkeit

1. Eine TM *akzeptiert* eine Eingabe $w \in \Sigma^*$, wenn sie nach Lesen von w im akzeptierenden Zustand stoppt.
2. Sie *akzeptiert* eine Sprache $L \subseteq \Sigma^*$, wenn sie genau die Wörter w aus L als Eingabe akzeptiert.
3. Eine Sprache $L \subseteq \Sigma^*$ heißt *rekursiv* oder *entscheidbar*, wenn es eine Turingmaschine gibt, die auf allen Eingaben stoppt und ein Wort $w \in \Sigma^*$ genau dann akzeptiert, wenn $w \in L$ gilt.

Definitionen zur Entscheidbarkeit

4. Eine Sprache $L \subseteq \Sigma^*$ heißt *rekursiv-aufzählbar* oder *semi-entscheidbar*, wenn es eine Turingmaschine gibt, die ein Wort $w \in \Sigma^*$ genau dann akzeptiert, wenn $w \in L$ gilt.
   Das Verhalten der Turingmaschine für Eingaben $w \notin L$ ist damit nicht definiert. Sie stoppt entweder nicht in einem Endzustand oder aber stoppt gar nicht.
5. Eine TM *realisiert* die Funktion $f : \Sigma^* \to \Gamma^*$ mit

$$f(w) = \begin{cases} \text{Ausgabe der TM nach Abarbeitung von } w & \text{wenn die TM hält} \\ \text{undefiniert} & \text{sonst} \end{cases}$$

Nicht-Entscheidbarkeit

Beispiel: Diagonalsprache $L_D = \{w \in \Sigma^* \mid M_w \text{ akzeptiert } w \text{ nicht }\}$

- Annahme L_D entscheidbar
- $\Rightarrow \exists$ Turingmaschine T, die L_D entscheidet
- $\Rightarrow \exists t$ Gödelnummer von T
- $t \in L_D$?
- $\Rightarrow$ Widerspruch

Halteproblem

. Das Halteproblem beschreibt die Aufgabe zu entscheiden, ob eine Turingmaschine bei gegeben Eingabewort hält oder nicht. Dieses Problem ist im allgemeinen Fall semi-entscheidbar, aber nicht entscheidbar.

- Formal: $(\langle M \rangle, w) \in HALT \Leftrightarrow M$ hält bei der Eingabe w

Beispielaufgabe Entscheidbarkeit (B5 A4)

Zeigen Sie, dass die Sprache
$\mathcal{L} = \{\langle\mathcal{M}\rangle \mid \text{Turingmaschine } \mathcal{M} \text{ akzeptiert jede Eingabe}\}$
nicht entscheidbar ist!

Aufgabe B6 A1

Zeigen Sie, dass die Sprache
$\mathcal{L} =$
$\{\langle \mathcal{M} \rangle \mid \text{Turingmaschine } \mathcal{M} \text{ hat mindestens einen unerreichbaren Zustand}\}$
nicht entscheidbar ist!

Postsches Korrespondenzproblem

Gegeben sei eine Folge P von Paaren $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ von nichtleeren Worten über einem endlichen Alphabet. Dies nennt man eine **Instanz** des PKP. Eine nichtleere Folge $I = i_1, i_2, \ldots, i_m$ von Indizes aus $\{1, \ldots, n\}$ heißt Lösung zu P, wenn $x_{i_1} x_{i_2} \ldots x_{i_m} = y_{i_1} y_{i_2} \ldots y_{i_m}$.

Postsches Korrespondenzproblem

Gegeben sei eine Folge P von Paaren $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ von nichtleeren Worten über einem endlichen Alphabet. Dies nennt man eine **Instanz** des PKP. Eine nichtleere Folge $I = i_1, i_2, \ldots, i_m$ von Indizes aus $\{1, \ldots, n\}$ heißt Lösung zu P, wenn $x_{i_1} x_{i_2} \ldots x_{i_m} = y_{i_1} y_{i_2} \ldots y_{i_m}$.

Beispiel

$$\left(\binom{a}{aba}, \binom{ab}{bb}, \binom{baa}{aa}\right)$$

Postsches Korrespondenzproblem

Gegeben sei eine Folge P von Paaren $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ von nichtleeren Worten über einem endlichen Alphabet. Dies nennt man eine **Instanz** des PKP. Eine nichtleere Folge $I = i_1, i_2, \ldots, i_m$ von Indizes aus $\{1, \ldots, n\}$ heißt Lösung zu P, wenn $x_{i_1} x_{i_2} \ldots x_{i_m} = y_{i_1} y_{i_2} \ldots y_{i_m}$.

Beispiel

$$\left(\begin{pmatrix} a \\ aba \end{pmatrix}, \begin{pmatrix} ab \\ bb \end{pmatrix}, \begin{pmatrix} baa \\ aa \end{pmatrix}\right)$$

Lösung: $1, 3, 2, 3$

$$\begin{pmatrix} a \\ aba \end{pmatrix}, \begin{pmatrix} baa \\ aa \end{pmatrix}, \begin{pmatrix} ab \\ bb \end{pmatrix}, \begin{pmatrix} baa \\ aa \end{pmatrix}$$

Aufgabe B5 A2

Geben Sie, sofern möglich, je eine Lösung für die folgenden Post-Systeme an!
Begründen Sie gegebenenfalls, warum es keine Lösung geben kann!

1. $$\left\{ \begin{pmatrix} aa \\ a \end{pmatrix}, \begin{pmatrix} b \\ aa \end{pmatrix}, \begin{pmatrix} a \\ aab \end{pmatrix} \right\}$$

2. $$\left\{ \begin{pmatrix} 01 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 101 \end{pmatrix}, \begin{pmatrix} 101 \\ 0 \end{pmatrix} \right\}$$

Aufgabe

Finde eine Lösung für folgende Instanz des PKP:

$$((001, 0), (01, 011), (01, 101), (10, 001))$$

Aufgabe

Finde eine Lösung für folgende Instanz des PKP:

$$((001, 0), (01, 011), (01, 101), (10, 001))$$

Eine kürzeste Lösung hat mindestens die Länge 66, z.B:

$$I_1 = (2, 4, 3, 4, 4, 2, 1, 2, 4, 3, 4, 3, 4, 4, 3, 4, 4, 2, 1, 4,$$
$$4, 2, 1, 3, 4, 1, 1, 3, 4, 4, 4, 2, 1, 2, 1, 1, 1, 3, 4, 3, 4, 1, 2,$$
$$1, 4, 4, 2, 1, 4, 1, 1, 3, 4, 1, 1, 3, 1, 1, 3, 1, 2, 1, 4, 1, 1, 3)$$

Aufgabe

Finde eine Lösung für folgende Instanz des PKP:

$$((001, 0), (01, 011), (01, 101), (10, 001))$$

Eine kürzeste Lösung hat mindestens die Länge 66, z.B:

$$I_1 = (2, 4, 3, 4, 4, 2, 1, 2, 4, 3, 4, 3, 4, 4, 3, 4, 4, 2, 1, 4,\\ 4, 2, 1, 3, 4, 1, 1, 3, 4, 4, 4, 2, 1, 2, 1, 1, 1, 3, 4, 3, 4, 1, 2,\\ 1, 4, 4, 2, 1, 4, 1, 1, 3, 4, 1, 1, 3, 1, 1, 3, 1, 2, 1, 4, 1, 1, 3)$$

Aufgabe B6 A3 rekursiv aufzählbare Mengen

Welche der folgenden Mengen sind rekursiv aufzählbar?
Beweisen Sie Ihre Aussage!

1. $M_1 := \{q \in \mathbb{Q} \mid 0 < q < 1\}$
2. $M_2 := \{r \in \mathbb{R} \mid 0 < r < 1\}$

Bis zum nächsten Mal!

Lizenzen