הרצאה 6

קוד האפמן

הקדמה

רוצים לשמור קובץ טקסט על הדיסק בצורה חסכונית. אפשרות אחת היא לקודד כל תו בטקסט במספר סיביות קבוע. מספר הסיביות שנזדקק לכל תו הוא $\lceil \log |\Sigma| \rceil$. אפשרות נוספת היא לקודד כל תו במספר סיביות שונה. נשים לב שקידוד כזה יכול להיות חסכוני יותר כאשר יש שוני בין שכיחויות התווים בטקסט.
דוגמה:
עבור הא"ב $\{A, B, C, D\}$ והמחרוזת הבאה: AAABCD קידוד באורך קבוע יהיה באורך $6 \times 2 = 12$.

לעומת זאת, אם נבחר את הקידוד הבא

A	1
B	01
C	001
D	000

אז אורך הקידוד יהיה 11 בלבד.

הגדרה 1 (קוד בינרי)**.** בהינתן א"ב סופי Σ קידוד הוא פונקציה שממפה כל תו בא"ב למחרוזת בינרית $c : \Sigma \to \{0,1\}*$

הגדרה 2 (הרחבה של קוד)**.** הרחבה של קוד היא פונקציה $c : \Sigma^* \to \{0,1\}$ שמוגדרת להיות $c(t_1 \dots t_k) = c(t_1) \dots c(t_k)$

תכונות

נבחן שלושה קידודים שונים לא"ב $\{A, B, C, D\}$

$$c_1 = \begin{array}{|c|l|} A & 1 \\ B & 01 \\ C & 001 \\ D & 000 \end{array} \quad c_2 = \begin{array}{|c|l|} A & 0 \\ B & 01 \\ C & 011 \\ D & 111 \end{array} \quad c_3 = \begin{array}{|c|l|} A & 1 \\ B & 01 \\ C & 011 \\ D & 111 \end{array}$$

באופן טבעי נדרוש שהקוד יהיה ניתן לפענוח (חד פענח), כלומר נרצה שההרחבה תהיה פונקציה חד חד ערכית.
דוגמה: ניתן לפענח את c_1, ו-c_2, אבל לא את c_3.
תכונה רצויה היא שנוכל לפענח כל תו ברגע שקראנו את המילה שמקודדת אותו (פענוח מידי).
דוגמה: התכונה מתקיימת עבור c_1, אבל לא מתקיימת עבור c_2, ו-c_3.

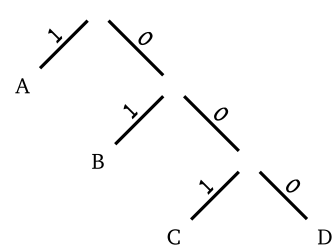
קודים חסרי רישות

קוד c יקרא חסר רישות אם לא קיימים $a, b \in \Sigma$ כך ש-$c(a)$ רישא של $c(b)$
קל לראות שקודים חסרי רישות ניתנים לפענוח וכן לפענוח מידי. מעבר לכך המשפט הבא (ללא הוכחה) מראה שלמטרתנו מספיק להתמקד בקודים חסרי רישות.

משפט 1. לכל קוד חד פענח c קיים קוד חסר רישות c' כך שלכל $a \in \Sigma$ מתקיים $|c(a)| = |c'(a)|$.

קוד חסר רישות כעץ בינרי

ניתן לייצג כל קוד חסר רישות כעץ בינרי, למשל את הקוד c_1 ניתן לייצג על ידי העץ הבא:



נשים לב שבתיאור כזה ישנה התאמה חד חד ערכית בין עלי העץ למילות קוד.

קוד האפמן

נניח שנתון לנו קובץ טקסט מעל א"ב Σ, וכן נתונה לנו פונקציה שמתארת את מספר המופעים של כל תו בקובץ $f : \Sigma \to \mathbb{N}$
נרצה למצוא קוד חסר רישות (עץ בינרי) שיקודד את הקובץ במינימום סיביות, כלומר:

$$\min_c \sum_{a \in \Sigma} |c(a)| \cdot f(a)$$

במונחים של עצים נרצה למצוא עץ שממזער את הערך

$$\min_c \sum_{a \in \Sigma} d(a) \cdot f(a)$$

כאשר $d(a)$ הוא עומק העלה שמתאים לתו a בעץ.
לעץ שממזער את הערך הנ"ל נקרא עץ האפמן

טענה 1. כל עץ האפמן הוא עץ מלא (לכל צומת פנימי יש שני בנים)

הוכחה. נסתכל על עץ האפמן שממזער את מספר הצמתים הפנימיים עם בן אחד, נניח בשלילה שיש בן כזה אז אפשר להחליף צומת כזה עם הבן שלו ולהקטין את ערך העץ $\square$

טענה 2. אם $a, b \in \Sigma$ שני איברים בעלי ערך f מינימלי, אז קיים עץ האפמן שבו a ו-b הם אחים ובעלי עומק מקסימלי

הוכחה. אם לא, נבחר שני עלים אחים בעלי עומק מקסימלי ונחליף אותם עם a ו-b. $\square$

למה 1. אם $a, b \in \Sigma$ שני איברים בעלי ערך f מינימלי, נגדיר $\Sigma' = \Sigma \setminus \{a, b\} \cup \{z\}$ כאשר $z \notin \Sigma$. כמו כן נגדיר $f(z) = f(a) + f(b)$.
אם T' עץ האפמן של Σ' אז העץ T שמתקבל מ-T' על ידי החלפה של העלה z בצומת פנימי עם שני בנים a ו-b הוא עץ האפמן של Σ.

הוכחה. ניקח עץ האפמן $\hat{T}$ על Σ שבו a ו-b אחים. ממנו נייצר עץ $\hat{T}'$ על Σ' על ידי איחוד העלים a ו-b לעלה z. נראה שמתקיים

$$w(T) = w(T') + f(a) + f(b) \leq w(\hat{T}') + f(a) + f(b) = w(\hat{T})$$

$\square$

אלגוריתם לבניית עץ האפמן

1. אם $|\Sigma| = 2$ מחזירים עץ בינארי עם 3 צמתים
2. יהיו $a, b \in \Sigma$ שני האיברים עם ערכי f מינימליים
   (א) מגדירים $\Sigma' \leftarrow \Sigma \setminus \{a, b\} \cup \{z\}$
   (ב) קובעים $f(z) = f(a) + f(b)$
   (ג) קוראים לאלגוריתם באופן רקורסיבי על Σ' ומקבלים T' מוסיפים לעלה z ב-T' את הבנים a ו-b לקבלת T
   (ד) מחזירים T

טענה 3. האלגוריתם מחזיר עץ האפמן

הוכחה. באינדוקציה על גודל הא"ב ובעזרת למה 1 $\square$

דוגמת הרצה: גנן גידל דגן בגן