קורס: קודים לתיקון שגיאות ושימושיהם במדעי המחשב
מרצה: ד״ר קלים יפרמנקו
סמסטר: סתיו תשפ״א
תאריך: 29/12/2020

<u>הרצאה 19</u>

<u>List – decoding RS</u>

<u>מבוא:</u>

בחלק הראשון נעשה חזרה על list decoding Reed Solomon מעצם חשיבותו, לאחר מכן נראה כיצד להכליל את שיטת reed Solomon לקודים אחרים, בהם אפשר לקבל תוצאות <u>כמעט אופטימליות</u>

פיתוח שיטת קידוד יעילה וכן התפתחות תקשורת הנתונים הובילו לשימוש נרחב בקודי-תיקון שגיאות ובראשם בקודי RS. בימינו ניתן למצוא מערכות אלקטרוניות רבות אשר עושות שימוש בקודי RS, דוגמת מכשירי רדיו (הן קוויים, דוגמת ערוצי DSL, והן אלחוטיים דוגמת WiMAX), תקליטורים, DVD ומקודדי-וידאו. שימושים נוספים של הקוד הם לשיפור מערכות פיזור ספקטרום של מערכות תקשורת אלחוטיות, וכן בהתמודדות עם שגיאות חד-כיווניות, מהסוג הנפוץ באפיקי נתונים מהירים של מחשבים.

ניזכר באלגוריתם RS שלמדנו:

אלגוריתם RS מקבל כ**קלט** n נקודות (α_i, y_i) ופרמטר k המהווה דרגת הפולינום, ומחזיר כ**פלט** את כל הפולינומים $f(x)$ מדרגה לכל היותר k כך ש $\{|i| : f(\alpha_i) = y_i \geq 2\sqrt{nk}\}$
כלומר מספר המקומות הנכונים הוא לפחות $2\sqrt{nk}$.

<u>תיאור האלגוריתם:</u>

- למצוא פולינום $0 \not\equiv Q(x,y)$ כך ש $Q(\alpha_i, y_i) = 0$ ומתקיים:
  $\deg_y Q(x,y) \leq \sqrt{\frac{n}{k}} = d_y$ (הדרגה המקסימלית בה - y מופיע)
  $\deg_x Q(x,y) \leq \sqrt{nk} = d_x$ (הדרגה המקסימלית בה - x מופיע)
  כלומר – $Q(x,y) = \sum_{i=1}^{d_y} f_i(x) y^i : \deg(f_i(x)) \leq d_x$
- למצוא את כל הפולינומים $p(x)$ כך ש $Q(x,p(x)) \equiv 0$, אשר דיברנו בשיעור על כך שקיימים אלגוריתם למציאת $p(x)$ ובנוסף שמספרם של הפולינומים המקיימים זאת לא גדול.

<u>הנקודות החשובות שאנו צריכים לבדוק על מנת להבטיח את נכונות האלגוריתם הן:</u>

- אכן קיים $Q(x,y)$ השונה מפולינום ה-0 ומתקיימים התנאים שדרשנו.
  נבחין כי אוסף כל הפולינומים $Q(x,y)$ המקיימים $\deg_x Q(x,y) \leq d_x$, $\deg_y Q(x,y) \leq d_y$ הינם מרחב לינארי - ניתן לראות על ידי סגירות לחיבור וכפל בסקלר, מכך:
  <u>$Q(\alpha_i, y_i) = 0$ זה תנאי לינארי ולכן מתקיים בהכרח</u>

$v = \{Q(x,y): \deg_x Q \le d_x \ \wedge \deg_y Q \le d_y\}$ - מרחב הפולינומים, נחשב את מימד המרחב:

$$Q(x,y) = \sum_{i=0}^{d_x}\sum_{j=0}^{d_y} c_{ij}x^i y^j$$

כל פולינום נקבע באופן יחיד על ידי אוסף המקדמים, ואורך וקטור המקדמים הינו:
$\dim v = (d_x + 1) * (d_y + 1)$.

אם $\dim v > n$ כלומר אם יש יותר משתנים ממשוואות במערכת לינארית הומוגנית,
אז קיים פתרון שהוא לא 0. המשוואה שמגדיר הפתרון היא:

$$Q(\alpha_k, y_k) = 0 \iff \sum_{i=0}^{d_x}\sum_{j=0}^{d_y} c_{ij}\alpha_k^i y_k^j = 0$$

קיימות n משוואות כאלו, ומספר הנעלמים (c_{ij}) יותר גדול מ n, לכן קיים פתרון השונה מ-0 למשוואות הלינאריות.

- נבחין כי כאשר $Q(x,y)$ הינו פולינום האפס, אכן מתקיימת הדרישה $Q(\alpha_i, y_i) = 0$.
אם כך מדוע לדרוש ש $Q(x,y)$ יהיה שונה מפולינום האפס?

על פי האלגוריתם, הפלט הינו כל הפולינומים מדרגה $\le k$ כך ש $Q(x, p(x)) \equiv 0$.
לכן אילו $Q(x,y)$ הינו פולינום האפס, אזי **כל** $p(x)$ מקיים ש $Q(x,y)$ הינו פולינום האפס,
ואילו $Q(x,y) \not\equiv 0$ אז יש לכל היותר $\deg_y Q(x,y)$ פולינומים המקיימים.

- אם $p(x)$ מקיים שעבור $2\sqrt{nk}$, $p(\alpha_i) = y_i$ אז $Q(x, p(x)) \equiv 0$

הבעיה בשלב השני באלגוריתם אינה אלגוריתמית, אלא להוכיח שהתשובה הנכונה תהיה ברשומה שלנו, כלומר שהפולינום שמקיים את תנאי הסף שלנו, באמת יקיים את התנאים.
על מנת להראות זאת סימנו $F(x) = Q(x, p(x))$ והוכחנו כי ל$F(x)$ יש יותר שורשים מהדרגה, ולכן הוא פולינום האפס.
לכל i כך ש $p(\alpha_i) = y_i$ מתקיים $F(\alpha_i) = Q(\alpha_i, p(\alpha_i)) = Q(\alpha_i, y_i) = 0$

ומכאן שלכל i המקיים $p(\alpha_i) = y_i$ אז $F(\alpha_i) = 0$. לכן מספר ה α_i בהם הקוד אינו משובש שווה לפחות למספר השורשים בפולינום $F(x)$.
אנו יודעים שמספר ה- i-ים הוא לפחות $2\sqrt{nk}$.

נותר לדעת מה דרגת $F(x)$:

מהגדרת $F(x)$ $\Leftarrow$ $F(x) = Q(x, P(x))$ ומהגדרת $Q(x)$ $\Leftarrow$ $Q(x, P(x)) = \sum_{i=1}^{d_y} f_i(x) y^i$

לכן $F(x) = \sum_{i=0}^{d_y} f_i(x) p(x)^i$

קורס: קודים לתיקון שגיאות ושימושיהם במדעי המחשב
מרצה: ד"ר קלים יפרמנקו
סמסטר: סתיו תשפ"א
תאריך: 29/12/2020

ניזכר כי $\deg p(x) \le k$, מכאן $\deg p(x)^i \le ki$

$$\deg F(X) = \max\{\text{דרגות המונומיים}\} =$$
$$= \max\{\deg f_i(x)p(x)^i\} \le \max\{\deg f_i + ik\} \le \max\{d_x + ik\}$$

$d_x + ik$ מקסימלי עבור $i = d_y$. ולכן $\deg F(X) \le d_x + kd_y = \sqrt{mk} + k\sqrt{\frac{n}{k}} = 2\sqrt{nk}$

קיבלנו כי ל$F(x)$ יש יותר שורשים מהדרגה, ולכן הוא פולינום האפס, כנדרש.

אפשר לשפר את האלגוריתם אם כאשר אנו מחפשים את $Q(x,y)$ אנחנו דורשים ש $\deg f_i \le d - ki$
עד כה דרגות ה- f_i היו זהות, אנחנו מעוניינים לאזן את דרגות המונומים.
נבחין כי גם אילו היינו מגדילים את $\deg f_0(x)$ להיות kdx עדיין זה לא היה מעלה את דרגת הפולינום $F(x)$.

<u>שיפורים לאלגוריתם:</u>

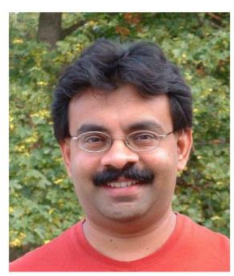
1. אם נבחר את f_i כך ש $\deg f_i \le D - ik$ (עבור $D = \sqrt{2nk}$)
כלומר במקום לקבוע את דרגת f_i להיות d_x, עכשיו אנו קובעים לכל f_i דרגה אחרת, ונקבל אלגוריתם טוב יותר המפענח אם יש רק $\sqrt{2nk}$ נקודות נכונות (לעומת $2\sqrt{nk}$)

אפשר לשפר גם ל $\sqrt{nk}$: עד כה בנינו פולינום $Q(x,y)$ שמתאפס בנקודות (α_i, y_i) ובאמצעות זאת, אילו ידענו שיש מספר נקודות שהוא נכון, אז היינו יכולים לוודא שלפולינום $F(x)$ יש הרבה שורשים.

הרעיון הוא שבמקום שלפלונים $Q(x,y)$ יהיה 0 אחד בנקודה (α_i, y_i), אפשר לדאוג שיתאפס הרבה פעמים בנקודה זו. לדוגמא, לפולינום $Q(0,0)$ יש t אפסים אם המקדם החופשי של c_{00} הוא 0.
ו $k-$ אפסים אם לכל i,j כך ש $i+j \le t$ מתקיים $c_{ij} = 0$. נראה אלגוריתם העושה זאת

<u>**הרחבה - אלגוריתם גורוסוואמי-סודן** – המשפר ל $\sqrt{nk}$</u>

בהינתן קוד ריד סולומון (n,k)בעל הנקודות $(\alpha_1, \dots, \alpha_n)$ ומספר חיובי r,
האלגוריתם מקבל וקטור $\beta = (\beta_1, \dots, \beta_n)$ ומחזיר פולינומים מדרגה $\le k$ המתאימים למילות הקוד באופן חד חד ערכי.


סודן


גורוסוואמי

קורס: קודים לתיקון שגיאות ושימושיהם במדעי המחשב
מרצה: ד"ר קלים יפרמנקו
סמסטר: סתיו תשפ"א
תאריך: 29/12/2020

הגדרה: לפולינום $Q(x,y)$ יש r שורשים בנקודה (α,β) אם ל- $Q(x,y)$ קיימים r ריבוי אפסים בנקודה (α,β) כאשר $(\alpha,\beta)=(0,0)$

<u>תיאור האלגוריתם:</u>

נגדיר את מילות הקוד המועברות להיות $(f(\alpha_1),\dots,f(\alpha_n))$ עבור $(\alpha_1,..,\alpha_n)$ בהתאמה, והמילים שהתקבלו הן $(\beta_1,\dots,\beta_n)$

- עבור וקטור $(\beta_1,\dots,\beta_n)$, נבנה פולינום עם שני משתנים, $Q(x,y)$ עם $(1,k)$ דרגות ממושקלות של לכל היותר d כך שעבור Q קיימים r ריבוי אפסים בנקודות (α_i,β_i) לכל $1\le i\le n$ $Q(\alpha_i,\beta_i)$
- מצא את הגורמים של $Q(x,y)$ מהצורה $y-p(x)$ וגם $p(\alpha_i)=\beta_i$ עבור לפחות t ערכים של i כאשר $0\le i\le n$ וגם $p(x)$ הוא פולינום מדרגה $\ge k$, כאשר נזכור כי הפולינומים מדרגה $\ge k$, מתאימים למילות הקוד באופן חד חד ערכי.

<u>הוכחת נכונות:</u>

נגדיר

$$Q(x,y)=\sum_{i=0,j=0}^{i=m,j=p}\alpha_{i,j}x^iy^j$$

כאשר $\deg Q(x,y)=m$ וגם $\deg_y Q(x,y)=p$ לכן

$$Q(x+\alpha,y+\beta)=\sum_{u=0,v=0}^{r}Q_{u,v}(\alpha,\beta)x^uy^v \quad * \text{ בהמשך נוכיח}$$

כאשר

$$Q_{u,v}(x,y)=\sum_{i=0,j=0}^{i=m,j=p}\binom{i}{u}\binom{j}{v}\alpha_{i,j}x^{i-u}y^{j-v}$$

הוכחת *:

$$Q(x+\alpha,y+\beta)=\sum_{i,j}\alpha_{i,j}(x+\alpha)^i(y+\beta)^j$$

לפי הבינום של ניוטון

$$Q(x+\alpha,y+\beta)=\sum_{i,j}\alpha_{i,j}(\sum_u\binom{i}{u}x^u\alpha^{i-u})(\sum_v\binom{i}{v}y^v\beta^{j-v})$$

$$Q(x+\alpha,y+\beta)=\sum_{u,v}x^uy^v(\sum_{i,j}\binom{i}{u}\binom{i}{v}\alpha_{i,j}\alpha^{i-u}\beta^{j-v})$$

$$Q(x+\alpha,y+\beta)=\sum_{u,v}Q_{u,v}(\alpha,\beta)x^uy^v$$ כנדרש

<u>טענה:</u> שלב האינטרפולציה משרה $\binom{r+1}{2}$ אילוצים על מקדמי α_i.

<u>הוכחה:</u> לפולינום $Q(x,y)$ קיימים r ריבוי אפסים בנקודה (α,β) כאשר $Q_{u,v}(\alpha,\beta)=(0,0)$ עבור $0\le u+v\le r-1$

קיימים $r-v$ ערכים עבורם u מקיים את האי שוויון, מכאן שמספר האילוצים הוא $\sum_{v=0}^{r-1} r-v=\binom{r+1}{2}$

לכן, $\binom{r+1}{2}$ אילוצים יכולים להיות עבור (u,v), וכל בחירה מהווה אילוץ על מקדמי α_i כנדרש.

כיוון ש $y-p(x)$ הוא גורם של $Q(x,y)$, ניתן לייצג את $Q(x,y)$ באופן הבא:

$Q(x,y)=L(x,y)\big(y-p(x)\big)+R(x)$ כאשר $L(x,y)$ היא מנה המתקבלת כאשר מחלקים את $Q(x,y)$ ב $Q(x,y)$ ו- $R(x)$ היא השארית. כעת, אם נחליף את y ב$p(x)$, $Q\big(x,p(x)\big)\equiv 0$ רק אם $R(x)\equiv 0$

<u>טענה:</u> אם $p(\alpha)=\beta$ אז $(x-\alpha)^r$ גורם של $Q\big(x,p(x)\big)$

<u>הוכחה:</u>

כפי שהוכחנו $Q(x,y)=\sum_{u,v} Q_{u,v}(\alpha,\beta)(x-\alpha)^u(y-\beta)^v$

$$Q\big(x,p(x)\big)=\sum_{u,v} Q_{u,v}(\alpha,\beta)(x-\alpha)^u(p(x)-\beta)^v$$

בהינתן $p(\alpha)=\beta (p(x)-\beta) mod(x-\alpha)=0$ אז $(x-\alpha)^u(p(x)-\beta)^v mod(x-\alpha)^{u+v}=0$

לכן $(x-\alpha)^r$ הינו גורם של $Q\big(x,p(x)\big)$.

כפי שהוכחנו, $t.r>D$ לכן $t>\frac{D}{r}$

$\frac{D(D+2)}{2(k-1)}>n\binom{r+1}{2}$ כאשר צד שמאל של אי שוויון הינו חסם עליון על מספר המקדמים של $Q(x,y)$ וצד ימין הוכח בטענה.

$D=\sqrt{knr(r-1)}$ לכן, $t=\left[\sqrt{kn(1-\frac{1}{r})}\right]$, כאשר $r=2kn$ אזי $t>\left[\sqrt{kn(1-\frac{1}{2})}\right]>\left[\sqrt{kn}\right]$

<u>קיבלנו כי ניתן לפענח עבור $\sqrt{nk}<$ נקודות נכונות, כנדרש</u>

<u>שיתוף סוד-secrete sharing:</u>

בקריפטוגרפיה, Secret sharing היא בעיה של פיצולו של סוד בין קבוצת שותפים, באופן שאינו ידוע לאף אחד מהם לחוד וניתן לגלותו רק באמצעות שיתוף פעולה של כל או חלק מחברי הקבוצה.
הסוד בהקשר של מדעי המחשב הוא ערך מספרי כלשהו ויכול להיות בעל חשיבות קריפטוגרפית כגון מפתח הצפנה או סיסמה.

קורס: קודים לתיקון שגיאות ושימושיהם במדעי המחשב
מרצה: ד"ר קלים יפרמנקו
סמסטר: סתיו תשפ"א
תאריך: 29/12/2020

דוגמא: יש k חברים והם מעוניינים לנעול את כספם כך שרק אם כולם מתאספים אז ניתן לפתוח את המנעול.
בצורה פורמאלית: בהינתן k אנשים וסוד שנסמנו s (סיסמא), רוצים לתת לכל אחד חלק מהסוד כך שרק כולם יחד יכולים לפענח את s.

לדוגמא עבור k = 4 ו- s = password:

$pa \Leftarrow 1$

$ss \Leftarrow 2$

$wo \Leftarrow 3$

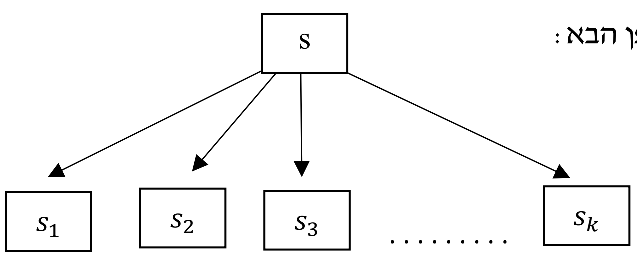
$rd \Leftarrow 4$

למרות שקיימים 26^8 אפשריות לסיסמא מאורך 8 מעל א'ב'-אנגלית, אם שלושת האנשים 1,2,3 מתאספים, הם יודעים ש-s מהצורה: passwo _ _, (באופן דומה עד כדי מיקום האותיות החסרות עבור כל שלושה אנשים), עליהם לבדוק רק 26^2 קומבינציות לגילו s ולפיענוח הסיסמא.

<u>המטרה של secrete sharing היא:</u>

1. לתת לכל אחד חלק s_i כך שאם k-1 אנשים מתאספים אין להם מושג מהי הסיסמא, כלומר כל $|א'ב'|^{|s|}$ האפשרויות ל-s להיות הסיסמא הן אפשרויות באותה מידה.
2. אם k אנשים מתאספים אז הם יכולים לגלות את s

נרצה לחלק את $s \in \{0,1\}^n$ באופן הבא:
כך שתעמוד בתנאים שהגדרנו.



נגריל מספרים $r_1, r_2, .., r_{k-1} \in \{0,1\}^n$.
האדם ה- i עבור $1 \le i \le k-1$ מקבל את r_i, והאדם ה-k מקבל $s \oplus r_1 \oplus r_2 \oplus ... \oplus r_{k-1}$
נבחין כי אם $k-1$ האנשים הראשונים מתאספים, כל שידוע להם זה מספרים אקראיים ללא קשר ל-s.
בנוסף, כאשר מבצעים $s \oplus r$, עבור: $r =$ ביטים אקראיים, מה שנקבל זה אוסף של ביטים אקראיים ולא נוכל להבדיל בין $s \oplus r$ ל- r מספר אקראי. כלומר לכל x $pr(s \oplus r = x) = 2^{-n}$ לכן אפילו אם מתאספים k-1 אנשים, הם יכולים לגלות רק $s \oplus r_{i1} \oplus r_{i2} ... \oplus r_{i(k-1)}$
עבור $\{i1, .., i(k-1)\} \subseteq \{1, .., k\}$ אשר עבורם הינם רק אוסף ביטים אקראיים. אך אם k האנשים מתאספים, אם הם מבצעים xor בין הסודות שלהם, ערך התוצאה שווה s.