ממן 14 - אורי זית

1. נציג מכונה שמכריעה את SUBSET SUM עם שימוש ב-O(n) מקום:

- עבור קלט : <S,t> כאשר S={x1,...,xn}

- נגדיר משתנה sum

- עבור כל m תת קבוצה של S :

א - רשום את m על הסרט(במקום m הקודמת)

ב - sum=0

ג - סכום את איברי m במשתנה sum

ד - אם הם שווים ל-t קבל

- דחה

נכונות - אנחנו בודקים לכל תת-קבוצה אם הסכום שלה שווה ל-t לכן אם קיימת תת-קבוצה כזו בהכרח נמצא אותה באחת האיטרציות ואם לא קיימת נגיע לדחיה.

סיבוכיות - אנחנו משתמשים בכל איטרציה במקום כדי לאחסן את m הנוכחית שהיא חלקית לS כלומר O(n) ובנוסף אנחנו משתמשים במשתנה sum כאשר בכל איטרציה הוא יכול להגיע לכל היותר לסכום של כל איברי S - O(n)

2. א,ב,ג- כן לכל הסעיפים, הוכחה - נראה אלגוריתם :

- נחזיק משתנה counter מאותחל ל-0

- עבור כל השמה לנוסחה שבקלט :

- רשום את ההשמה הנוכחית(במקום הקודמת)

- בדוק אם היא מספקת את הנוסחה, אם כן העלה את counter באחד

- בסעיף א- בדוק אם counter>=k אם כן קבל אחרת דחה, בב' - בדוק אם counter<=k אם כן קבל אחרת דחה, בג'- בדוק אם counter=k אם כן קבל אחרת דחה.

נכונות- אנחנו בודקים כמה השמות מספקות בדיוק יש עבור הנוסחה ולכן גם כמה לכל היותר/לכל הפחות/בדיוק.

סיבוכיות- אנחנו שומרים בכל פעם השמה אחת בלבד- O(n)

ובנוסף את counter שתופס לכל היותר log(2n)=n.(2n זה מספר ההשמות האפשריות)

3. א' - נשתמש במכונה F המוגדרת בהוכחת משפט 4.5 ,

המכונה מכריעה את השפה וסיבוכיות המקום היא O(N2) כי המצבים באוטומט שמתקבל בבניה הם מכפלה קרטזית של מצבי A וB.

ב' - נוכיח ש- ונקבל לפי 8.5 ש- .

נסמן : מספר המצבים ב-A ו-B ב- m ו-k בהתאמה.

מקורס אוטומטים - קיימים אסד"ים B',A' שקולים ל A,B עם לא יותר מ-2K,2M מצבים.

**אם** היינו בונים אס"ד C מ-A',B' כמו ב4.5 אז מספר המצבים ב-C היה חסום ע"י 2k+m .

לכן אם C לא ריקה קיימת בה מילה שאינה ארוכה מ-2k+m (מתקבלת במסלול ללא מעגלים)

ומילה זו (ע"פ הבניה) קיימת בהכרח רק באחד מ A,B

לכן נבנה מכונה שבודקת באופן לא דטרמניסטי אם יש מילה שמתקבלת רק באחד מהאסלד"ים כך -

- נבחר אסל"ד שבו המילה מתקבלת - בה"כ A.

- נאתחל counter=0

- נבחר בכל פעם אות אחת(במקום האות שהייתה) שהיא תהיה האות הבאה במילה

- עבור A מבין המצבים אליהן האות הנוכחית מעבירה אותו נבחר מצב אחד ונשמור אותו.

- עבור B נשמור את קבוצת המצבים אליהן הוא עובר בקריאת האות הנוכחית

- נבדוק - אם A במצב מקבל ו-B בקבוצת מצבים שכולם לא מקבלים - נקבל

- counter++

- if(counter==2k+m) דחה.

סיבוכיות המקום של המכונה המצבים שנשמרים בכל שלב+האות - O(n)

המונה - log(2k+m)<=log(2n)=n.

4. כן, הוכחה: קיימת מ"ט שמכריעה את C שבשאלה בסיבוכיות מקום פולינומיאלית

נגדיר את E: נרוץ על Σ\* בסדר הסטנדרטי כך :

הרץ את המ"ט על המילה הנוכחית

אם המילה התקבלה הדפס אותה

נקה את הסרט מלבד המילה הנוכחית

ועבור למילה הבאה ע"י הוספת/החלפת אות .

קל לראות שהמונה מפיק את C ,

סיבוכיות מקום - לכל מילה w עד הדפסתה המילים שהודפסו לפניה תופסות מקום לינארי(כי לא מחקנו אותן) ועבור כל מילה המ"ט השתמשה במקום פולינומיאלי באורך המילה שהייתה שהוא קטן או שווה לאורך w .

5. טענת עזר : אם שפה A NL שלמה אז גם המשלימה שלה NL שלמה - הוכחה(אלעזר בפורום):

בגלל שוויון המחלקות, גם המשלימה בNL  
לכל שפה B ב NL יש רדוקציה לA  
בגלל שוויון המחלקות, גם למשלימה של B יש רדוקציה ל-A.  
הרדוקציה הזו היא גם רדוקציה של B למשלימה של A.  
לכן, גם המשלימה של A היא NL-שלמה.

נראה שהמשלימה (v שייך למעגל מכוון) בNL ושיש אליה רדוקציה מ- PATH בכך נקבל שהמשלימה היא NL שלמה ולכן גם D:

**D** ∈ **NL:**

נשתמש באותו אלגוריתם מדוגמה 8.19 רק שבמקום לבדוק אם הגענו אל t (ִצומת היעד ב-PATH)

נבדוק כל פעם אם הגענו ל-v.

**נראה כעת רדוקציה מ-PATH אל המשלימה :**

נשתמש ברדוקציה שבתשובה 8.23 בספר רק שבמקום להחזיר את <G> נחזיר את <G,t> ואז מובטח לנו שיש מעגל ש-t נמצא בו א"םם יש מסלול מt ל-s (כאשר <G,s,t> זה הקלט של PATH).

6. נראה שהשפה ב-L ויחד עם ההנחה בשאלה נקבל שהשפה לא ב-NL-COMPLETE

כי אז היינו מקבלים שNL⊆L ומכך גם NL=L בסתירה להנחה.

אלגוריתם :

- נחזיק סרט עבודה עם 21 מקומות כאשר כל מקום בו הוא בגודל log(n) כך שהוא יכול לייצג כל צומת בגרף(הצומת ה-n צריך log(n) סיביות כדי לייצגו)

- נעבור על כל הצמתים ועבור כל צומת נריץ dfs כאשר בכל הוספת צומת למסלול ששומרים בריצת dfs נבדוק אם הגענו ל21 צמתים ואם כן -נקבל, אם סיימנו את ההרצות על כל הצמתים ולא מצאנו - נדחה.

נכונות נובעת מנכונות dfs

סיבוכיות מקום- 21log(n)=O(log(n)) ויחד עם זה שהאלגוריתם דטרמניסטי נקבל שהשפה ב-L.