**מבחן ב"מודלים חישוביים " 372.1.2306**

**מועד א' 25/06/14**

מרצה: **פרופ' אריאל פלנר**

מתרגל: **מר גוני שרון**

משך המבחן: שעתיים וחצי

יש לענות על שאלות 1 ו2.

יש לענות על שלוש שאלות מתוך 3-6

ניקוד ינתן לכל שאלה כמכלול ולא לפי הסעיפים.

**(שאלה 1, 28 נקודות)**

בעית הסוכן הנוסע Traveling Salesman Problem (TSP) היא:

בהינתן

א) גרף משוקלל

ב) קודקוד התחלה

ג) מספר K

האם קיים מסלול שיוצא מקודקוד ההתחלה , עובר בכל הקודקודים השייכים ל- פעם אחת בלבד וחוזר לקודקוד ההתחלה במסלול שמשקלו הוא K.

רמז: ידוע ש .

1. בעיית הסוכן הבררן (PSP) היא: בהינתן
2. גרף משוקלל
3. סט קודקודים
4. קודקוד התחלה
5. מספר K

האם קיים מסלול שיוצא מקודקוד ההתחלה , עובר בכל הקודקודים השייכים ל- פעם אחת בלבד וחוזר לקודקוד ההתחלה במסלול שמשקלו הוא K.

הוכיחו במדוייק שבעיית הסוכן הבררן שייכת ל NPC

1. בהינתן פתרון לבעיה נוודא שהפתרון עובר פעם אחת בכל קודקודי S ושמשקלו שווה K סה"כ O(|V|)
2. נבצע רדוקציה מ TSP
3. בהינתן בעיית TSP(G,v,K) ניצור בעיית PTP(G',S,v',K') כך ש: G'=G, S=V, v'=v, K'=K
4. בעיית TSP זהה לחלוטין לבעיית PTP הנוצרת בעזרת הטרנספורמציה (זו אותה הבעיה).
5. הטרנספורמציה לא כוללת שום שינוי (בהנחה שלא עושים deep copy) סה"כ O(1)
6. בעיית הסוכן המטייל (WSP) היא: בהינתן
7. גרף משוקלל
8. סט צלעות
9. קודקוד התחלה
10. מספר K

האם קיים מסלול שיוצא מקודקוד ההתחלה , עובר בכל הצלעות מתוך פעם אחת בלבד וחוזר לקודקוד ההתחלה במסלול שמשקלו הוא K.

הוכיחו שגם בעית הסוכן המטייל שייכת לNPC. כאן צריך רק להראות את הרדוקציה (את הבניה).

נבצע רדוקציה מ-PTP. בהינתן בעיית PTP(G,S,v,K) ניצור בעיית WSP(G',S',v',K') חדשה כך ש: v'=v, K'=K, גרף G' יהיה זהה לגרף G בתוספת קשת עצמית במשקל 0 מכל קודקוד השייך ל-S לעצמו. S' יכיל את כל הקשתות העצמיות החדשות שיצרנו.

**(שאלה 2, 12 נקודות, 3 לכל סעיף נכון.)**

נניח כי קיים אלגוריתם דטרמיניסטי הפותר את בעיית SAT בזמן ריצה של . כלומר, גם החסם העליון וגם החסם התחותן הוא . עבור כל אחת מהמסקנות הבאות נא להקיף בעיגול האם היא נכונה או לא.

|  |
| --- |
| 1. ייתכן ש P=NP - לא נכון |
| 1. ניתן לפתור כל בעיה ב NP, בזמן ריצה - נכון |  |
| 1. לכל בעיהR ב NP קיים קבוע CR , כך שהחסם העליון לפתרון הבעיה הוא O(nCR) - לא נכון 2. לא קיימת בעיה ב NPC שיכולה להיפטר בזמן פולינומיאלי. - נכון |  |

**(שאלה 3, 20 נקודות)**

1. נתונה השפה:

כלומר מספר הb שונה ממספר הa כפול 2.

אם L רגולרית הראו ביטוי רגולרי או אוטומט סופי מתאים. אם היא לא הוכיחו שלא.

לא רגולרית

נניח שכן ולכן גם השפה המשלימה שלה רגולרית .

מכאן שקיים אוטומט סופי בעל m מצבים המקבל את

נבחר מילה w=a^(m)b^(2m)

w שייכת ל L

|w|>=m

נפרק את המילה לשלוש:

X=a^i

Y=a^j : 1<=j<=m, i+j<=m

Z=a^(k)b^(2m)

i+j+k=m

האוטומט יקבל כל מילה מהסוג a^(i+j\*t+k)b^(2m) : t>=0

עבור t=2 נקבל a^(i+2j+k)b^(2m)=a^(m+j)b^(2m)

כיוון ש- 1<= j אז בהכרח 2(m+j) ≠ 2m

ולכן הגענו למילה שאיננה בשפה משמע לא קיים כזה אוטומט. השפה המשלימה אינה רגולרית וכך גם השפה המקורית.

**(שאלה 4, 20 נקודות)**

נתונה השפה

1. האם L רגולרית? אם כן תנו ביטוי רגולרי מתאים, אם לא ציינו מילה שניפוחה פעמיים לפי עקרונות למת הניפוח יוביל למילה שאינה שייכת לשפה.

רגולרית

1. תנו דקדוק חסר הקשר המתאים לשפה L.

כללי גזירה:

S->RbR/aRbaR

R->aa/ **λ**

**(שאלה 5, 20 נקודות)**

סרטטו מכונת טיורינג המקבלת מילים מהשפה

ראו פתרון בתרגול 9

**(שאלה 6, 20 נקודות)**

נסחו את בעית העצירה (HALT). הוכיחו שהיא אינה ניתנת לפתרון.

ההוכחה לאי-כריעות בעיית העצירה נעשית בדרך השלילה: מראים כי מההנחה שניתן לפתור את בעיית העצירה מגיעים לסתירה - קיימת תוכנית שעבור קלט מסוים חייבת גם לעצור וגם לא לעצור לעולם.

נניח שיש אלגוריתם ( Halt(Q,X שמכריע בשאלה האם Q עוצרת על X, ונתבונן בתוכנית A הבאה המקבלת תוכנית Q כקלט:

אם Halt(Q,Q), הכנס ללולאה אינסופית.

אחרת, עצור.

נשים לב כי מעבירים כאן את Q פעמיים: הן בתור התוכנית שיש לבדוק והן בתור הקלט לתוכנית שעליו היא נבדקת. אין בעיה עקרונית בהעברת Q כקלט - ניתן לחשוב על כל קלט כרצף של תווים, ולכן גם תוכנית מחשב יכולה להיחשב לקלט.

נשאל עכשיו, האם תעצור A עבור הקלט A? (כלומר, במקרה שהתוכנית A תקבל את עצמה) נחלק לשני מקרים, ונקבל סתירה בשניהם:

נניח ש- A תעצור. מכיוון שעל פי הצורה שבה הגדרנו את A, היא עוצרת רק אם לא מתקיים (Halt(A,A, נסיק כי כך הם פני הדברים. אולם, מכיוון שמהגדרת האלגוריתם Halt הוא אינו מתקיים רק אם A אינה עוצרת על עצמה נגיע לסתירה - הנחנו ש- A עוצרת וקיבלנו שהיא בהכרח אינה עוצרת.

כעת נניח כי מתקיים ההפך: A נכנסת ללולאה אינסופית. על פי הגדרת A, זה קורה רק אם מתקיים (Halt(A,A. לכן, מהגדרת האלגוריתם Halt, נובע ש A עוצרת עבור הקלט A - כלומר, הנחנו שהיא אינה עוצרת וקיבלנו כי היא בהכרח עוצרת.

הנחנו שקיים אלגוריתם הפותר את בעיית העצירה והגענו לסתירה, לכן לא ייתכן אלגוריתם לפתרון בעיה זו.