**מבחן ב"מודלים חישוביים ואלגוריתמים" 372.1.2306**

**מועד א' 2/7/12**

מרצה: **ד"ר אריאל פלנר**

מתרגל: **מר גוני שרון**

משך המבחן: שלוש שעות

יש לענות על חלק א' בשלמותו.

בנוסף יש לענות על 4 מתוך 5 שאלות בחלק ב'.

ניתן לענות על שאלת הבונוס בנוסף לשאלות מחלק ב. (היא תשמש כמגן)

ניקוד יינתן לכל שאלה כמכלול ולא לפי הסעיפים.

**חלק א' (חובה) 32 נקודות**

(שאלה 1 , 24 נקודות)

1. הגדירו את המושג: פסוק לוגי מהצורה 4-CNF
2. הגדירו את בעיית ההכרעה 4-CNF-SAT

קלט: פסוק לוגי מהצורה 4-CNF

פלט: True אם קיימת לפחות השמה אחת למשתנים כך שהפסוק הנתון יהיה אמת. False אחרת.

1. הוכיחו הוכחה מלאה ש4-CNF-SAT שייכת ל NPC
2. **הבעיה ב :NP –**

כיוון ש SAT ב NP גם 4-CNF-SAT ב NP (כמקרה פרטי)

**2.** **בחירת בעיה:**

נבצע רדוקציה מבעיית 3-CNF-SAT.

**3. טרנספורמציה:**

נבחר שרירותית איבר אחד מכל פסוקית ונוסיף שיכפול שלו לפסוקית כך יתקבלו פסוקיות עם 4 ליטרלים.

1. **הוכחת נכונות:**

שיכפול ערך בפסוקית לא משנה את קבוצת ההשמות שהיא מקבלת. כיוון שהשיכפול והאיבר המקורי מקבלים השמה זהה השפעתם על נכונות הביטוי זהה. כיוון שקבוצת ההשמות האפשריות לכל פסוקית זהה גם קבוצת ההשמות של הביטוי המלא זהה.

1. **הטרנספורמציה פולינומיאלית:**

עבור כל פסוקית נוסיף ליטרל נוסף. סה"כ לינארי במספר הפסוקיות.

1. הוכיחו עבור המקרה הכללי שעבור K ≥ 3 מתקיים ש K-CNF-SAT שייכת לNPC . (כאן מספיקה טענה כללית על K, אבל לא צריך הוכחה מלאה הכוללת את כל חמשת השלבים.)

ההוכחה זהה להוכחה בסעיף הקודם אלא שכעת במקום להוסיף שיכפול אחד של אחד הליטרלים נוסיף K שיכפולים.

(שאלה 2, 8 נקודות)

נתונות 3 בעיות a,b,c כולם ב NP. קיימת רדוקציה פולינומיאלית מ-a ל-b ומ-b ל-c.

לפניכם ארבעה הגידים. עבור כל אחד רשמו האם הוא נכון או לא נכון. תשובה נכונה מזכה ב (2+), תשובה לא נכונה מורידה (2-). סעיף ללא תשובה יקבל אפס נקודות. תשובה נכונה אם היא בהכרח נכונה (תמיד נכונה). נא לכתוב במחברת את התשובות שבחרתם.

|  |  |
| --- | --- |
| 1. c ∈ NP-C 🡨🡪 a ∈ NP-C (אם ורק אם) | נכון לא נכון |
| 1. c ∈ P 🡪 a ∈ P | נכון לא נכון |
| 1. קיימת רדוקציה פולינומיאלית מ-a ל-c | נכון לא נכון |
| 1. c ∈ P 🡪 P = NP | נכון לא נכון |

**חלק ב' (68 נקודות)**

יש לענות בדיוק על 4 מתוך 5 השאלות 4-8 (17 נקודות לשאלה).

(שאלה 3, 17 נקודות)

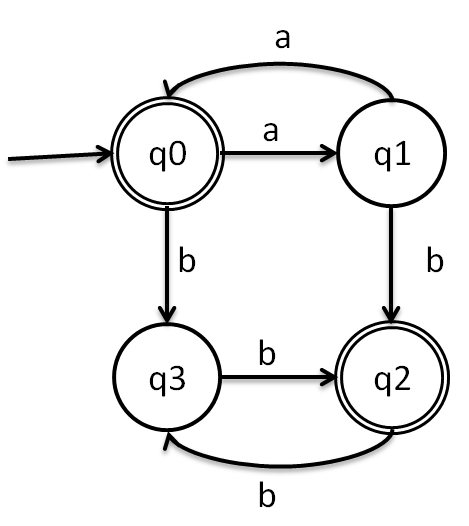
נתונה השפה  מעל הא"ב {a,b}

(כלומר מספר הa זוגי אם ורק אם מספר הb זוגי)

האם שפה זו רגולרית?

אם השפה רגולרית הראו אוטומט (דטרמנסטי או לא דטרמנסטי) או ביטוי רגולרי עבורה. אם השפה אינה רגולרית – הוכיחו.

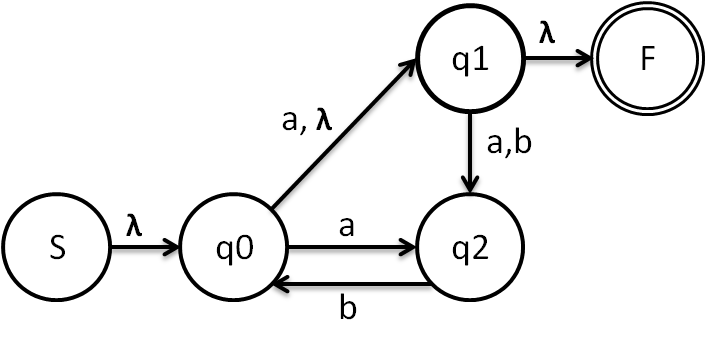
השפה רגולרית לפי הביטוי: aa\*bb\*+a(aa)\*b(bb)\*

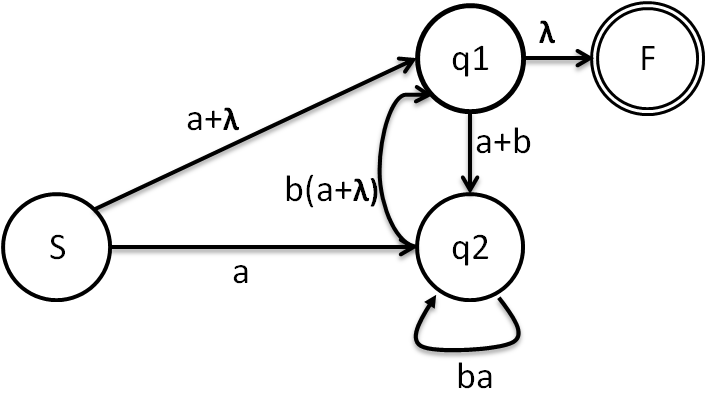
או לפי האוטומט הבא:

(שאלה 4, 17 נקודות)

נתון האוטומטNFA שבציור. הראו שלב אחרי שלב כיצד הופכים אוטומט זה לביטוי רגולרי. בכל שלב, יש לבטל קודקוד. ביטול הקודקודים חייב להתבצע לפי סדר המספרים המציינים את הקודקודים.

מהו הביטוי הרגולרי שהתקבל?





(שאלה 5, 17 נקודות)

הוכיחו שמכונת טיורינג אינסופית משני צדדים שקולה למכונת טירוינג שהיא אינסופית מצד אחד

הדרכה: יש להוכיח את שני הכיוונים.

(בכיוון הקשה צריך להראות (לצייר) אילו מצבים התווספו לאוטומט)

כפי שהוצג בתרגול.

(שאלה 6 , 17 נקודות)

1. נתון הגרף הממושקל הבא. צריך למצוא מסלול קצר ביותר מS ל T.

E3

E2

E1

1. תנו משקלות לצלעות כך שהאלגוריתם BFS לא יחזיר פתרון אופטימלי אך אלגוריתם דייקסטרה כן יחזיר פתרון אופטימלי.

E1=3, E2=1, E3=1

1. תנו משקלות לצלעות כך שאלגוריתם דייקסטרה לא יחזיר פתרון אופטימלי.

E1=2, E2=3, E3=-2

1. כיצד ניתן לזהות מעגלים שליליים בעזרת אלגוריתם Floyd-Warshall?

אמ"ם מופיע ערך שלילי באלכסון המרכזי של המטריצת המוחזרת אז יש מעגל שלילי מהקודקוד אותו מייצג הערך.

(שאלה 7 , 17 נקודות)

בעיית All pairs shortest paths(K) מוגדרת כדלהלן:

נתונה מטריצת משקלות L1=W (עבור גרף עם n קודקודים). אנו רוצים לחשב מטריצה שבה יהיו המרחקים הקצרים ביותר כאשר בכל מסלול מותר לכל היותר K צלעות.

1. רשמו את הפונקציה extend() (כפי שלמדנו), הגדירו את הקלט ואת הפלט שלה.

הוגדר בהרצאה.

1. רשמו פונקציה המשתמשת בextend() ופותרת את הבעיה שניתנה.

L=W

For i 🡨 0 until k

L=extend(L,W)

Return L

1. מה זמן הריצה כפונקציה של n וK

K\*n\*n\*n=O(K\*n^3)

1. רשמו אלגוריתם המשתמש ברעיון של כפל מטריצות גם עבור בעיה זו.

L=W

For i 🡨 0 until log(k)

L=extend(L,L)

Return L

\* נכון כאשר K הוא חזקה של 2 (כפי שהוגדר בזמן בחינה)

עבור המקרה הכללי: (בונוס)

L' = W

L = W

K' = 1

K'' = 1

While(k' != k)

L' = W

K'' = 1

While(k' + k'' <= k)

L = extens(L,L')

L' = extend(L',L')

k' = k' + k''

k'' = k'' + k''

Return L

1. מה זמן הריצה שלו כפונקציה של n וK

Log(K)\*n\*n\*n=O(Log(K)\*n^3)

(שאלה 8 רשות בונוס , 20 נקודות)

שאלה זו תוכל להחליף כל שאלה אחרת מחלק ב. כלומר אם ענינם על 4 שאלות מחלק ב וגם על שאלה זו, ייבחרו ארבע התשובות עם הניקוד הגבוהה ביותר.

1. השוו בקצרה בין DA לUCS עבור הקריטריונים הבאים:
2. Applicability

DA – חייב לקבל תיאור מלא של כל הגרף

UCS – מקבל קודקוד יחיד. לא צריך את תיאור הגרף.

1. דרישות הזכרון במהלך הריצה

DA – שמירת כל הקודקודים בגרף בכל מהלך הריצה.

UCS – שמירת הקודקודים שגולו אך כל שכניהם טרם גולו.

1. פעולות על תור העדיפויות במהלך הריצה

DA – כל קודקוד מוכנס לתור ומעודכן לפחות פעם אחת.

UCS – כל קודקוד נגיש מוכנס לתור פעם אחת אך לא בהכרח יעודכן.

1. השוו בין האלגוריתמים לגבי הפעולות על תור העדיפיות אם תור העדיפיות ממומש בעזרת רשימה ממוינת.

אם התור ממומש כרשימה ממוינת אין הבדל בין זמני הריצה בגלל שהוספת איברי \infty לא משפיעה על מיון האיברים בעל ההשמה.

1. Dijkstra כתב את מאמרו לפני שהמציאו מבנה נתונים יעיל לתור עדיפיות. מה ניתן לשער בעזרת התשובה שכתבתם בבסעיף ב. לגבי DA?

כיוון שלא היו הבדלים בזמני הריצה (במקרה הממוצע) DA התקבע בספרות. בעת שימוש בערמה איברי \infty כן משפיעים (כיוון שנאלץ לחלחל ערכים דרכם). לאחר פיתוח מבנה הערמה קיים יתרון ל USC במקרה הממוצע.