מרצה : פרופ׳ ח׳ אלדר פישר סמסטר חורף התשע״ב

מתרגלים : מר יבגני אברמוביץ׳
מר דימה אלנבוגן

מערכות מסדי נתונים
236363

פתרון

התיקונים (המעטים) שנעשו בזמן המבחן מסומנים באדום.

מועד ב׳ (1 באפריל 2012, ט׳ בניסן התשע״ב)

מס׳ ת.ז.

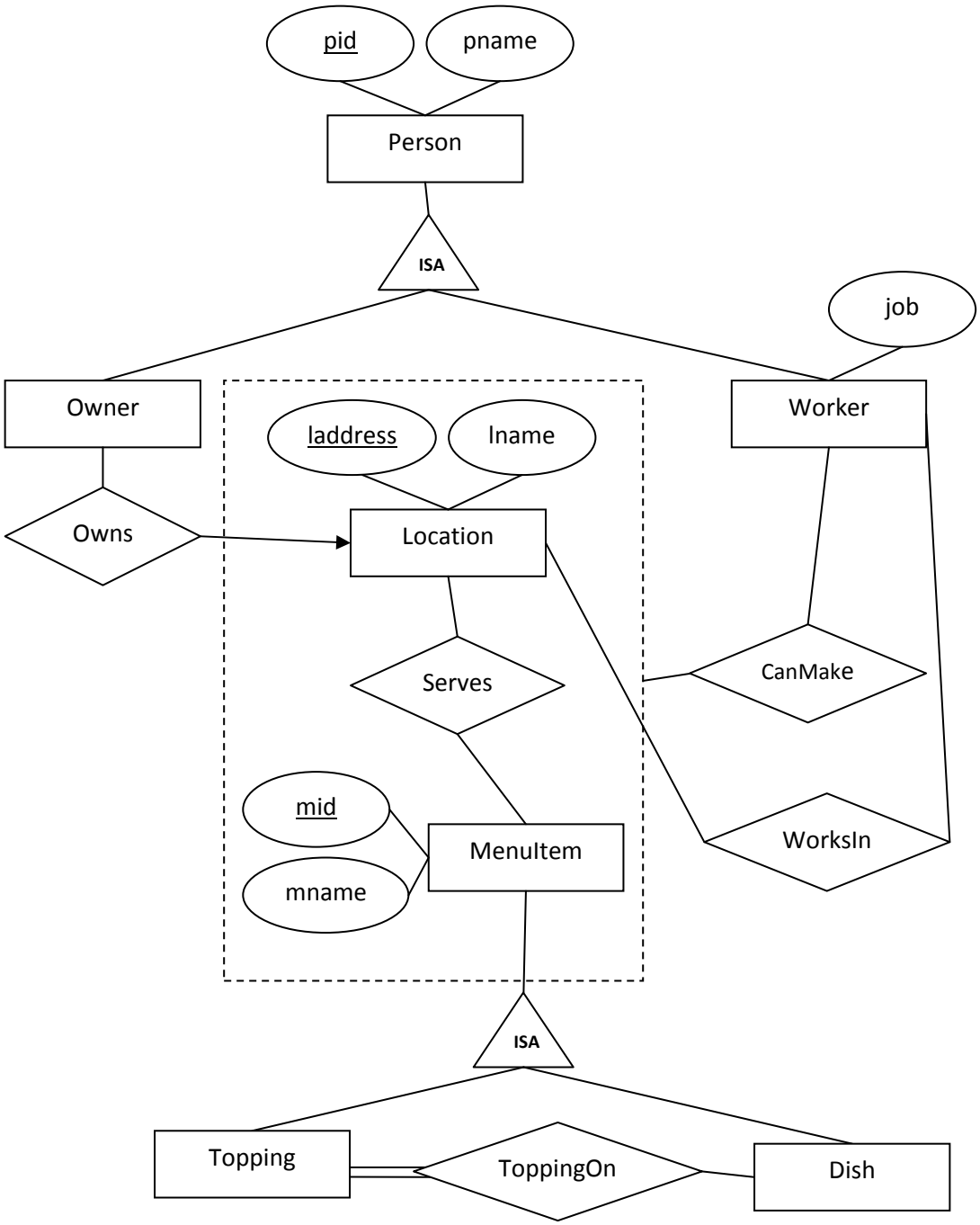
הנחיות לנבחנים

1. כתבו את התשובות אך ורק בטופס הבחינה, המחברת מיועדת לטיוטה בלבד.
2. מותר ומומלץ לכתוב את התשובות בעפרון.
3. בדף האחרון יש מקום נוסף לתשובות. אם צריך מקום נוסף לתשובות, השתמשו במקום זה תוך ציון הדבר ליד השאלה המקורית.
4. כל חומר עזר כתוב על נייר מותר בשימוש.
5. אין לקבל או להעביר חומר עזר כלשהו בזמן הבחינה.
6. בבחינה ארבע שאלות ללא בחירה. יש לענות עליהן במלואן.
7. בכל מקום שלא נאמר אחרת, יש לנמק את התשובות בקצרה. **תשובות לא מנומקות לא תתקבלנה, למעט במקומות שבהם אתם מתבקשים לכתוב שאילתה.**
8. יש להשתמש רק בסימנים או פונקציות שנלמדו בתרגול או בהרצאה או שמופיעות בשקפים של הקורס. כל שימוש בסימון שאינו כזה מחייב הסבר מלא של משמעות הסימון.
9. משך הבחינה שלוש שעות. תכננו את הזמן בהתאם. **לא תינתנה הארכות זמן במהלך המבחן.**
10. הבחינה (ללא דף הסריקה) כוללת 8 דפים, כולל דף זה (16 עמודים). נא לוודא שקיבלתם לידיכם את הטופס במלואו. **שימו לב שהבחינה מודפסת משני הצדדים.**
11. כאשר הניקוד של תתי הסעיפים אינו מצוין, ניקוד הסעיף מתחלק שווה ביניהם.
12. הניקוד אינו נועד לשקף את קושי השאלה ולכן מומלץ לקרוא קודם את כל השאלות.

בהצלחה!

שאלה 1 - ERD (29 נק')

נתונה דיאגרמה של רשת מזון מהיר:



הסבר לדיאגרמה:

יישויות:

- Person – אדם. לאדם יש מזהה pid ושם pname.
- Worker – עובד. לעובד יש תפקיד job.
- Owner – בעלים של סניף.
- Location – סניף. לסניף יש כתובת laddress ושם lname.
- MenuItem – פריט בתפריט. לפריט יש מזהה mid ושם mname.
- Dish – מנה עיקרית.
- Topping – תוספת.

קשרים:

- Owns – מציין בעלות על הסניף. אין תכונות נוספות בקשר זה.
- Serves – מציין שהסניף מגיש את הפריט. אין תכונות נוספות בקשר זה.
- CanMake – מציין שעובד יודע להכין את הפריט שמוגש בסניף. אין תכונות נוספות בקשר זה.
- ToppingOn – מציין תוספות אפשריות למנה. אין תכונות נוספות בקשר זה.

שאלות:

א. (6 נק') רוצים להוסיף את מחירי הפריטים לדיאגרמה כתכונה mprice.

i. איך מוסיפים את התכונה אם מעוניינים במחירים אחידים לכל הרשת?

מוסיפים את mprice ל-MenuItem.

ii. איך מוסיפים את התכונה אם מאפשרים לכל סניף לקבוע את המחיר?

מוסיפים את mprice ל-Serves.

ב. (8 נק') עבור השינויים הבאים, הסבירו את משמעותם. ענו על כל סעיף ביחס ל**דיאגרמה המקורית** ובאופן בלתי תלוי בסעיפים האחרים.

- תשובה שרק מציינת שם של הסימון החדש (כמו "X הופך להיות ישות חלשה") לא תתקבל. יש להסביר את השפעת השינוי על תכונות הישויות והקשרים בדיאגרמה.

i. הפיכת הקו בין Dish ל-ToppingOn לכפול.

לכל מנה חייבת להיות תוספת (לפחות אחת).

ii. הוספת קו התחתון תחת lname.

ייתכנו שני סניפים שונים בעלי אותה כתובת, אם יש להם שם שונה.

iii. פיצול משולש ה-ISA לשני משולשים, Worker יחובר לאחד מהם ו-Owner לשני.

בעל סניף יכול להיות גם עובד.

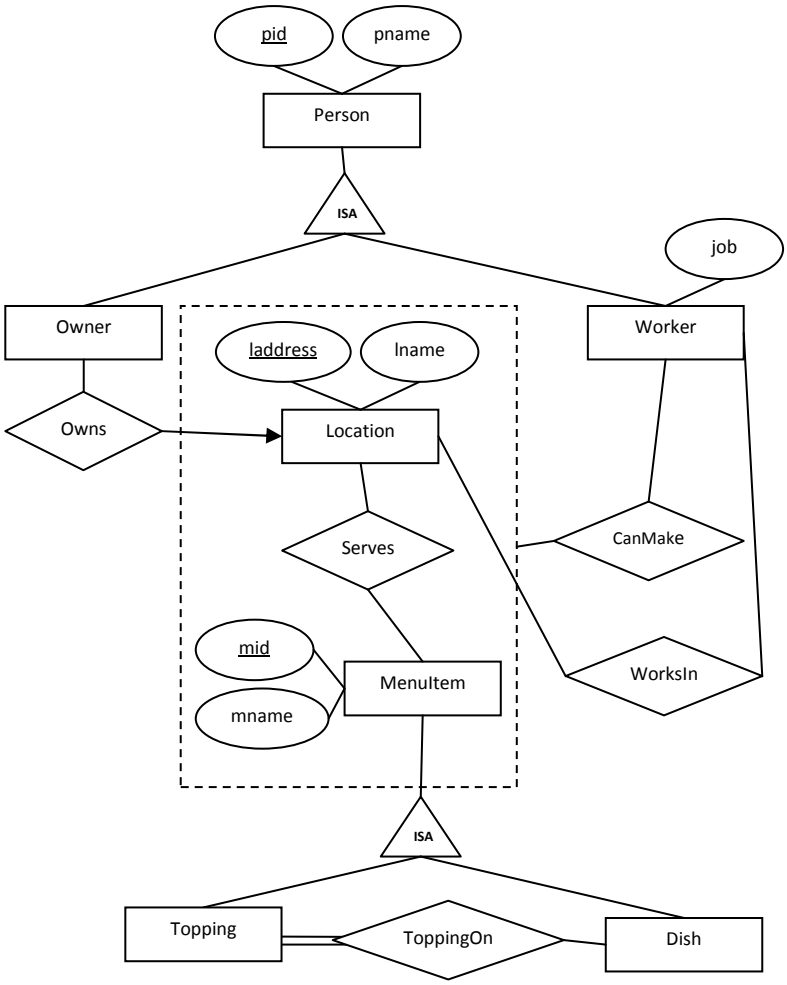
iv. הפיכת הקו בין Owns לOwner לחץ.

לכל סניף יש בעלים בודד (לכל היותר).

ג. (15 נק') לכל אחד מהתנאים הבאים שינו את הדיאגרמה בכדי שהוא יתקיים. ענו על כל תת-סעיף ביחס ל**דיאגרמה המקורית** ובאופן בלתי תלוי בסעיפים האחרים. אין להוסיף מגבלות מיותרות מעבר למה שצוין. בכל תת-סעיף נמצא העתק של הדיאגרמה המקורית שעליו יש לבצע את השינויים הנדרשים.

לחילופין, הנכם **רשאים** לתאר את השינוי במילים במלבן תחת הדיאגרמה.

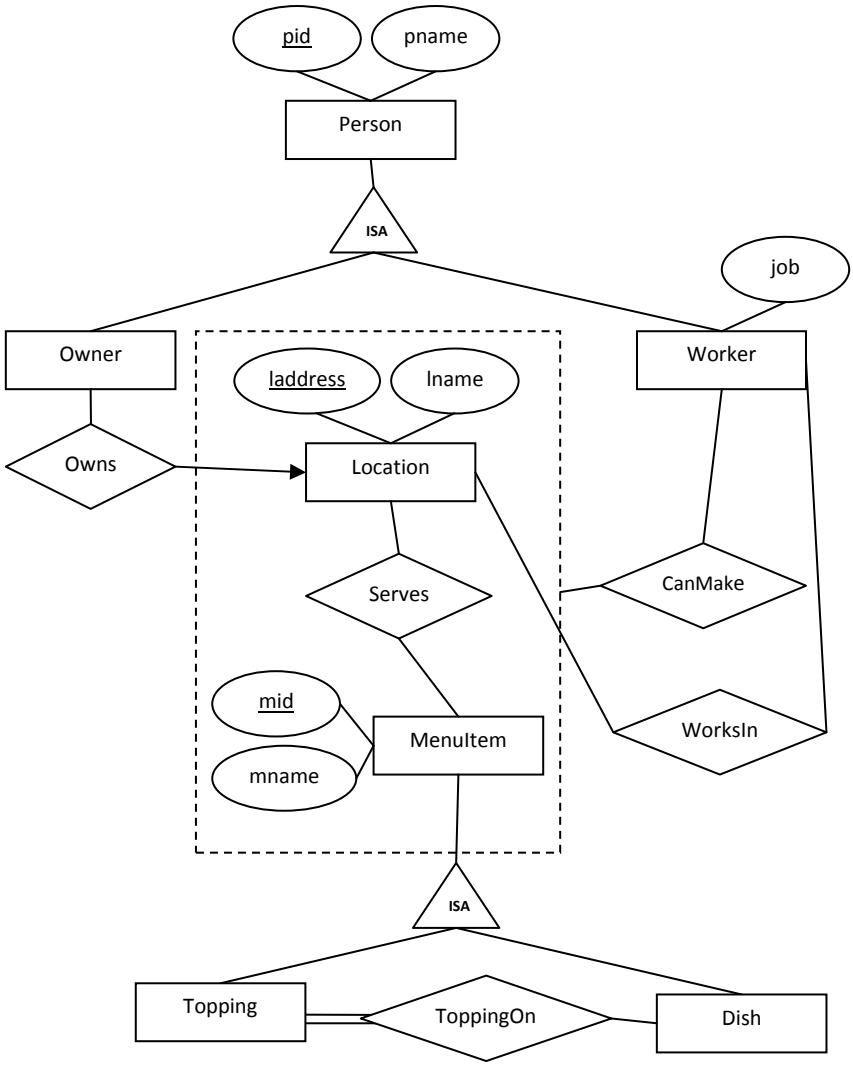
i. אם עובד יכול להכין פריט באיזשהו סניף, אז הוא יכול להכין אותו בכל הסניפים.



יש לחבר את CanMake ל-MenuItem במקום להקבצה סביב Serves.

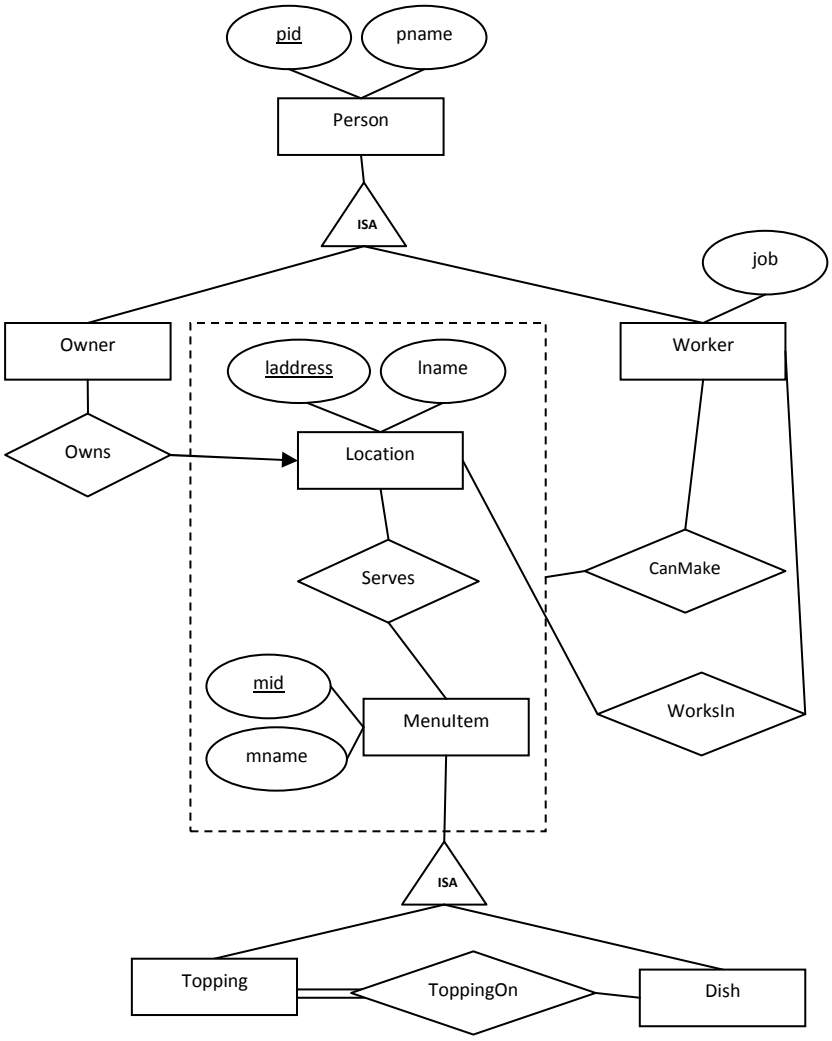
ii. ישנם גם מבקרי מסעדות. מבקר הוא אדם שאינו יכול להיות בעל סניף או עובד בו. המבקר יכול לתת ציון אחד לכל היותר למנה בסניף נתון בכל תאריך נתון.

"מנה" – הכוונה כאן היא לפריט בתפריט.



מוסיפים טיפוס ישות חדש critic ומחברים אותו לפיצול של person (שעתה יהיה פיצול לשלושה). מחברים אותו בטיפוס קשר חדש graded להקבצה של serves. לקשר החדש תהיה תכונת מפתח date ותכונת לא-מפתח grade.

iii. לא ייתכנו פריט ואדם כך שמזהה הפריט mid שווה למזהה האדם pid. מותר לשנות שמות של תכונות.



יש לחבר את Person ואת MenuItem בעזרת ISA לישות Identifiable שתכיל id, ולהסיר את pid ואת mid מטיפוסי הישות.

שאלה 2 – שאילתות מידע (28 נק')

נתונות הרלציות הבאות, אשר אינן בהכרח תואמות את ה-ERD מהשאלה הקודמת, ואינן בהכרח מהוות סכימה אופטימאלית עבור המסד:

location (address, name) – סניף בעל כתובת ייחודית address ושם name.

- כיסוי לכל התלויות הפונקציונאליות המתקיימות ב- location הוא {address→name}.

dish (id, name, locAddr, price) – מנה בעלת מזהה id ושם name מוגשת בסניף שכתובתו locAddr ומחירה price.

- כיסוי לכל התלויות הפונקציונאליות המתקיימות ב- dish הוא {id→name, id→price}.

א. (12 נק') נתונה הבעיה הבאה: בהינתן הטבלה dish, החזר קבוצת סניפים, המגישה את <u>כל המנות</u> המופיעות ב-dish. קבוצה זו צריכה להיות מינימאלית, כלומר אף תת קבוצה שלה אינה מגישה את כל המנות. (זוהי לא בהכרח קבוצת מינימום – תיתכן קבוצה מינימאלית אחרת קטנה יותר)
לפתרון הבעיה, נשתמש בדרך של יצירת טבלה של כל הסניפים, והסרת סניפים עד שאי אפשר להסיר יותר.

השלימו את תוכנית ה-C הנתונה כדי לממש את האלגוריתם וליצור את הטבלה כנדרש.
הניחו כי כל הפקודות מצליחות. בפרט, הניחו כי הטבלה שאתם מייצרים לא קיימת.

```
#include <libpq-fe.h>
#include <string.h>
#define FALSE 0
PGconn *conn;
PGresult *addresses = NULL, *countResult = NULL;
char *paramValues[2];
int paramLengths[2], binaryFormats[2] = {FALSE, FALSE};
int prevCount, count;
int main(void) {
      conn = PQconnectdb("host=pgsql.cs.technion.ac.il"
                         "dbname=fastfood user=aevgeny password=123456");
      /* ספור (בהשמטת כפילויות) את כל המנות המופיעות ב-dish */
      countResult = PQexec(conn, "
```

(2 נק')

```
SELECT COUNT(DISTINCT id) FROM dish;
```

```
      ");
      paramValues[1] = PQgetvalue(countResult,0,0);
      paramLengths[1] = strlen(paramValues[1]);

       /* צור טבלה שתכיל את כל הכתובות מ-locations */
      PQexec(conn, "
```

(2 נק')

```
CREATE TABLE Cover AS
  SELECT address FROM location;
```

```
      ");
      /* צמצם את הקבוצה עד אשר תגיע לגודלה המינימלי */
      prevCount = -1;
      count = 0;
      while (prevCount != count) {
            /* שלוף את הכתובות של קבוצת הסניפים הנוכחית */
            addresses = PQexec(conn, "
```

(2 נק')

```
SELECT address from Cover;
");
prevCount = count;
count = PQntuples(addresses);
for (i = 0; i < count; ++i) {
        /* מחק את הכתובת של סניף אם ניתן לצמצמו */
      paramValues[0] = PQgetvalue(addresses,i,0);
      paramLengths[0] = strlen(paramValues[0]);
      PQexecParams(conn, "
```

(6 נק')

```
DELETE FROM Cover WHERE
  address = '$1' AND
  $2 =
   (SELECT COUNT(DISTINCT id)
    FROM dish WHERE
     locAddr IN
      (SELECT address FROM Cover
       EXCEPT
       SELECT '$1');
```

קיים פתרון נכון ללא שימוש בפרמטרים:

```
DELETE FROM Cover C WHERE
  (SELECT COUNT (DISTINCT Id)
      FROM dish) =
   (SELECT COUNT(DISTINCT id)
    FROM dish WHERE
     locAddr <> C.address
```

```
      ",
      2, /* כמות הפרמטרים */
      NULL, /* טיפוסי הפרמטרים */
      paramValues, /* ערכי הפרמטרים */
      paramLengths, /* האורכים של הפרמטרים (בבתים) */
      binaryFormats, /*לכל פרמטר: האם נתון בצורה בינארית?*/
      FALSE); /* האם תוצאת הביטוי תישלף בצורה בינארית */
      }
      PQclear(addresses);
    }
    PQclear(countResult);
    PQfinish(conn);
    return 0;
}
```

ב. (8 נק') כתבו שאילתת RA המחזירה את שמות כל הסניפים, המגישים את כל המנות שמוגשות בשלושה סניפים לפחות. ניתן להגדיר רלציות עזר.

$$\pi_{name}(location \ltimes \rho_{locAddr \to address}$$

$$(\pi_{id,locAddr} dish \div \rho_{id1 \to id} (\pi_{id1}(\sigma_{\theta}(dish \times dish \times dish)))$$

לשם קריאות השתמשנו בבחירה בסימון "θ" עבור הביטוי:

$$\theta = (id1=id2) \wedge (id2=id3) \wedge (locAddr1 \neq locAddr2) \wedge$$

$$(locAddr2 \neq locAddr3) \wedge (locAddr1 \neq locAddr3)$$

ג. (8 נק') כתבו שאילתת DRC שתחזיר זוגות של מזהי סניפים, כך ששני הסניפים מגישים בדיוק את אותן המנות. יש לכתוב שאילתה שלא תהיה תלוית תחום (אין צורך להוכיח שהיא כזו).

$$\{(l1,l2) | \exists n1(location(l1,n1)) \wedge \exists n2(location(l1,n2)) \wedge$$

$$\forall did(\exists n \exists p(dish(did,n,l1,p)) \leftrightarrow$$

$$\exists n \exists p(dish(did,n,l2,p)))\}$$

שאלה 3 – תלויות פונקציונאליות (20 נק')

נתונה סכמה $R(A_1, A_2, \ldots, A_n)$, כאשר $n>2$. כמו כן נתונה קבוצת התלויות הפונקציונליות מעליה F.
ידוע שהתלויות הפונקציונליות ב-F הן <u>כל</u> התלויות מהצורה:

$\mathbf{A_iA_j \rightarrow A_{i+j}}$ בתנאי ש: $i + j \leq n$ וגם $i < j$.

דוגמא ל-R ו-F כאלה עבור $n=5$ נמצאת בסעיף ב'.

א. (8 נק') הוכיחו: F מינימאלית לכל n.

צריך להוכיח שני דברים.

ראשית – שמאף תלות אי אפשר למחוק תכונה מצד שמאל. מכיוון שמהתחלה כל התלויות הם משתי תכונות, הסגור של כל קבוצת תכונות בת תכונה אחת שווה לתכונה עצמה, ז"א שאין שום תלות לא טריביאלית מתכונה בודדת הנובעת מ-F. לכן אי אפשר למחוק תכונה בצד שמאל של אף תלות.

שנית – שאף תלות לא נובעת מהאחרות. אם נמחק את $\mathbf{A_iA_j \rightarrow A_{i+j}}$, ונסתכל על הסגור של $\mathbf{A_iA_j}$ לפי קבוצת התלויות הנותרת, נקבל שהוא שווה ל-$\mathbf{A_iA_j \rightarrow A_{i+j}}$ עצמה, כי לאף תלות אחרת צד שמאל אינו מוכל בקבוצה זו (יש תלות אחת בדיוק עבור $\mathbf{A_iA_j}$ כל כך ש-$i+j \leq n$ וגם $i<j$, ואין תלות ב-F מאף קבוצה מכילה ממש).

ב. (5 נק') נתון: $R(A_1, A_2, A_3, A_4, A_5)$, $F = \{A_1A_2 \rightarrow A_3, A_1A_3 \rightarrow A_4, A_1A_4 \rightarrow A_5, A_2A_3 \rightarrow A_5\}$
פרקו את R לכדי פירוק ρ מצורת 3NF. על ρ לשמר את המידע של R ואת התלויות של F.

לפי אלגוריתם הפירוק מהכתה (שימו לב שלפי סעיף קודם F היא כיסוי מינימלי):

R1 = {A1, A2, A3}

R2 = {A1, A3, A4}

R3 = {A1, A4, A5}

R4 = {A2, A3, A5}

{A1,A2} הינו מפתח קביל, ומוכל בסכמה R1. לכן אין צורך להוסיף עוד סכמות וזהו הפירוק הסופי.

ג. (7 נק') האם הפירוק שהתקבל בתת-הסעיף הקודם הוא BCNF? נמקו בקצרה.

לכל תת-סכמה R1…R4 ישנה רק תלות אחת לא טריביאלית שרלוונטית לה (בזכותה היא נוצרה ע"י האלגוריתם שלמדנו בכיתה). לכן אגף שמאל של התלות מהווה מפתח על באותה סכמה. זה מתקיים בכל תתי-הסכמות ולכן הפירוק הוא BCNF

שאלה 4 – XML (23 נק')

א. (5 נק') נתון קובץ DTD של מסמך filled.xml המתאר טופס עם ערכים ממולאים.

```
<!ELEMENT filled (#PCDATA|value)*>
<!ELEMENT value (#PCDATA)>
```

צומת המסמך הוא filled, ומתחתיו טקסט חופשי שבתוכו יש ערכים ממולאים (value).

כתבו תוכנית XPath 1.0 המוודאת האם אכן כל הערכים בקובץ filled.xml מולאו (בטקסט לא ריק). ציינו איך אתם מבדילים בין "כן" ו"לא". רצוי, אך לא חובה לכתוב ביטוי המחזיר ערך בוליאני, true/false.

```
not(/values/value[string-length(text())=0])
```

ב. (8 נק') נתונים שני קבצי XML, אחד של טופס ריק ואחד של רשימת ערכים.

קובץ form.xml עם צומת מסמך form המתאים ל-DTD הבא:

```
<!ELEMENT form (#PCDATA|entry)*>
<!ELEMENT entry EMPTY>
```

קובץ values.xml עם צומת מסמך values המתאים ל-DTD הבא:

```
<!ELEMENT values (value)*>
<!ELEMENT value (#PCDATA)>
```

ברצוננו למזג אותם לקובץ XML יחיד, ע"י הכנסת הערך (value) הראשון במקום המשבצת (entry) הראשונה, הערך השני במקום המשבצת השניה, וכו'. ניתן להניח שיש מספיק ערכים ב-values.xml למילוי כל הטופס של form.xml. על הפלט לציית ל-DTD של הסעיף הקודם.

הוצעה לצורך זה תוכנית ה-XQuery הבאה:

```
document {<filled>
{for $n in doc("form.xml")/form/node()
return if ($n/self::entry)
  then <value>doc("values.xml")/value[count($n/preceeding-sibling::entry)]</value>
  else $n}
</filled>}
```

תוכנית זו אינה עובדת כנדרש. מצאו ותקנו את **שלוש** השגיאות בה, שכולן נמצאות בשורת ה-then (השורה הנתונה בכתב עבה). לחילופין אפשר לכתוב שורה שתחליף שורה זו, ושעבורה התוכנית תעבוד כנדרש.

then ~~**<value>**~~doc("values.xml")/**values/**value[**1+**count($n/preceeding-sibling::entry)]~~**</value>**~~

התקבלה גם:

then <value>doc("values.xml")/**values/**value[**1+**count($n/preceeding-sibling::entry)]/**text()**</value>

למרות שהיא לא נכונה סינטקטית, וצריכה להיות:

then <value>**{**doc("values.xml")/**values/**value[**1+**count($n/preceeding-sibling::entry)]/**text()}**</value>

ג. (10 נק') חברה בשם Antiq עוסקת במכירת ספרים. להלן נתון ה-DTD של מסד הנתונים שלה (בשם ANTIQ.DTD):

```
<!ELEMENT antiq (book*)>
<!ELEMENT book (title, year, authors?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT authors (author+)>
<!ELEMENT author (#PCDATA)>
<!ATTLIST book ISBN ID #REQUIRED>
```

צומת המסמך הוא antiq. הניחו שאם מחבר (author) כתב או השתתף בכתיבת ספר הוא יופיע פעם אחת בדיוק תחת צומת הספר (book).

בחברה Antiq הוחלט להחזיק במלאי מִספָּר לא מוגבל של עותקים מכל ספר. עליכם להצמיד לכל עותק מזהה ייחודי כל שהוא (שונה מ-ISBN). כמו כן, עותקים אחדים נמצאו במצב רעוע וזקוקים לשחזור. לכן הוחלט להפרידם מרשימת העותקים למכירה **לרשימה נפרדת**.

התוכן המקורי של ANTIQ.DTD רשום במשבצת התשובה. עדכנו אותו כך שיתאפשר יישום של שתי ההחלטות הנ"ל. נא להימנע מביצוע שינויים מיותרים!

```
<!ELEMENT antiq (book*)>

<!ELEMENT book (title, year, authors?, forSale, inRestore)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT authors (author+)>

<!ELEMENT author (#PCDATA)>

<!ATTLIST book ISBN ID #REQUIRED>

<!ELEMENT forSale (copy*) >
<!ELEMENT inRestore (copy*) >
<!ELEMENT copy EMPTY>
<!ATTLIST copyId ID #REQUIRED>
```

קיבלנו גם תשובות שמימשו מתחת לצומת המסמך שלוש רשימות, אחת של ספרים, אחת של עותקים למכירה ואחת של עותקים לשיפוץ. במקרה זה על צומת העותק להכיל גם תכונת מצביע עבור הספר שהוא עותק שלו.

מקום נוסף לתשובות

אם אתם משתמשים בדף זה, ציינו זאת ליד השאלה/השאלות המקוריות, וציינו כאן את מספר/י השאלה/השאלות.

שאלה: _____ סעיף: _____

שאלה: _____ סעיף: _____