

## עבודת בונוס במערכות בסיסי נתונים:

### מציאת כל הכיסויים הקנוניים של קבוצת תלויות F.

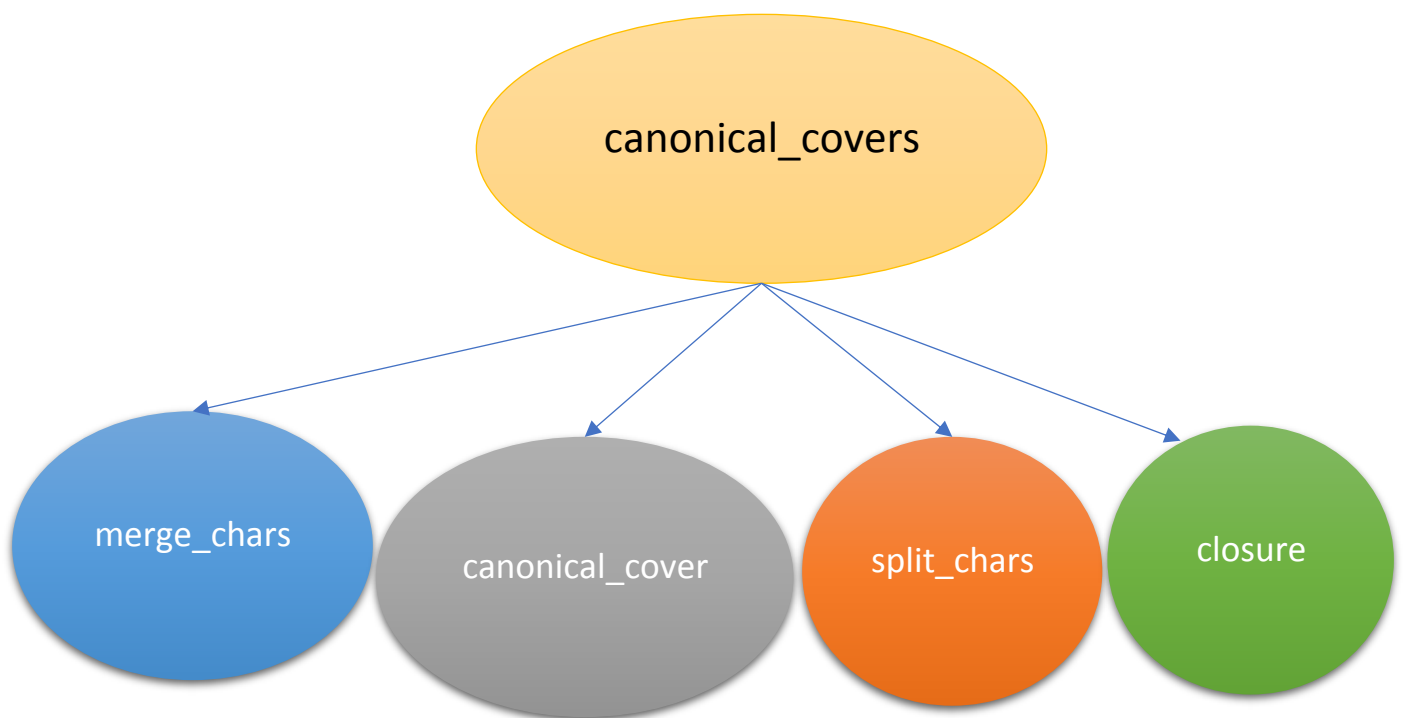
#### אמיר אברמוביץ ואייל פיקהולץ.

#### הקדמה:

אנחנו כתבנו את התוכנית בפרולוג.

בתחילת התוכנית רשמנו את הפונקציה closure שמחשבת את  $F +$  של קבוצת תלויות. נעזרנו בה על מנת למצוא את כלל הכיסויים הקנוניים של קבוצת התלויות F.

#### תיאור ע"י סכימה את התוכנית הראשית canonical covers בעבודה:



#### יתרון של פרולוג

כיוון שאנחנו עובדים עם פרולוג, הוא עובד בשיטת backtracking ולכן הוא יוכל למצוא את כל האפשרויות.

הסיבה: ניסיון לצדוק בשאליות / הסקות של מידע מסוים ואם לא מצליח הוא חוזר חזרה, מתקן לתכונה אחרת שכעת תוסק וימשיך בתהליך ההסקה. עד אשר יגיע לכל אפשרויות ההסקה.

#### מקורות נוספים

חקרנו באינטרנט אודות אלגוריתמים שונים בהם נוכל לפתח את האלגוריתם האידאלי לפתרון, ומצאנו מאמר מפורט שעל פיו אנו נעבוד על מנת למצוא את כלל הכיסויים הקנוניים.

אנו מחזירים את כל הכיסויים הקנוניים של קבוצת התלויות F כנדרש.

**.X**

**ב.**

λ

**.T**

## ה.

```

?- bonus_assignment:canonical_covers(abc, [a->b, b->c, c->a], Cover).
Correct to: "canonical_covers(abc,[[a->b],[b->c],[c->a]],Cover)"? yes
Cover = [[a->c], [b->a], [c->b]] ;
Cover = [[a->b], [a->c], [b->a], [c->a]] ;
Cover = [[a->b], [b->a], [b->c], [c->b]] ;
Cover = [[a->b], [b->c], [c->a]] ;
Cover = [[a->c], [b->c], [c->a], [c->b]] ;
false.

```

## פירוט לגבי הפונקציות :

א.

### closure (R,F,FPlus)

פונקציה זו מחשבת את קבוצת התכונות  $F +$  ע"י האלגוריתם הבא:

1. נעביר את  $R$  שהינו מחרוזת התכונות שקיבלנו כקלט לתוך רשימה כלומר  $abc$  יהפוך ל  $[a,b,c]$ .
  2. פירוק של שני צדדי קבוצת התכונות  $F$ , לצורה קנונית ( קבוצה רגילה עם פיסוק בין התכונות ).  
כלומר לדוגמא: מהקבוצה  $[a \rightarrow bf, b \rightarrow c, cjk \rightarrow a]$  לקבוצה:  $[(a \rightarrow [b, f]), ([b \rightarrow [c]), ([c, j, k] \rightarrow [a])]$ .
  3. פירוק של צד ימין מכל תכונה על מנת להשאיר תכונה בודדת בצד ימין ביחס. כלומר לדוגמא:  
 $([a \rightarrow [b, f])$  יהפוך לשני כללים:  $a \rightarrow b$ ,  $a \rightarrow f$ .
  4. ע"י הפונקציה `create_closure`, נחשב את קבוצת  $F +$  עבור הפלט של קבוצת היחסים שקיבלנו מתהליך 2 ו 3, עם קבוצת התכונות שקיבלנו מתהליך 1.
- נסביר בפירוט:

- א. בעזרת `powerSet` נחשב את קבוצת כל תתי הקבוצות של קבוצת התכונות שקיבלנו מתהליך 1.
- ב. בעזרת `without_trivial` נייצר את התכונות הנוספות ל  $F +$  ונמחק את כל התכונות הטריביות מהקבוצה  $F +$  שנחשב. על ידי חישוב של קבוצת התכונות  $\alpha +$  הניתנת להסקה ע"י  $F$  ועל ידי כל אחת מתתי הקבוצות שחישבנו בסעיף 4 א.
- ג. נדאג שצד ימין של היחסים יהיו בעלי תכונה אחת בלבד. זאת נעשה ע"י פירוק של היחסים.
4. נאחד בחזרה את  $F +$  מייצוג קנוני לייצוג של מחרוזת עבור קבוצת היחסים שקיבלנו.

ב.

### split\_chars (F0, F1)

לאחר שקיבלנו את  $F +$  עבור קבוצת התלויות  $F$ , נמיר אותה ע"י כך שנייצג אותה בייצוג קנוני.

ייצוג קנוני אומר לדוגמא: התכונה  $ab$  מומרת להיות רשימה של שתי תכונות כלומר:  $[a,b]$

ג.

### canonical cover(F1, F2)

- ניקח את ייצוג הקנוני של  $F +$  ונבצע את האלגוריתם הבא על מנת לקבל את הכיסוי הקנוני:
1. נאתחל משתנה שנקרא `minimal` שהוא יגדיר יחס בשפה של פרולוג לתכונה מינימלית.
  2. ניקח את קבוצת התלויות  $F +$  ונקרא לפונקציה אשר תפרק את צד ימין מכל תכונה על מנת להשאיר תכונה בודדת בכל צד ימין ביחס. פונקציה זו היא: `split_right`.
  3. ניקח את קבוצת התלויות  $F +$  מסעיף 2 לאחר פירוק של צד ימין.  
נקרא לפונקציה `minimalizeLeft` עם פרמטר של קבוצת תלויות זו והיא תפרק תלויות מרובות תכונות אשר מופיעות בצד שמאל בקבוצה.

יש להדגיש כי פונקציה זו איננה מפצלת באופן רגיל כמו שעושה הפונקציה `split_right`. אלא, היא בודקת עבור תלויות בקבוצה בהן הצד השמאלי מכיל יותר מתכונה אחת, האם ניתן להסיק את התלות ע"י הסרה של אחת התלויות. ע"י כך הפונקציה מצמצמת את הצד השמאלי בתלויות בקבוצה.

אופן פעולת פונקציית הצמצום – פונקציה שנקראת `non reducible` והיא חלק

מהפונקציה `minimalizeLeft`:

1. מסירה תכונה מתוך צד שמאל של התלות.
  2. מחשבת את קבוצת התכונות  $\alpha$  של צד שמאל החדש בתלות.
  3. בודקת האם קבוצה זו מכילה את צד ימין בתלות המקורית שאותה האלגוריתם רוצה לצמצם.
    - אם היא מכילה את צד ימין בתלות, נחזיר את התלות המעודכנת לאחר הצמצום.
    - אם היא לא מכילה את צד ימין בתלות, הרי כי התכונה שמחק האלגוריתם היא איננה מיותרת והאלגוריתם לא יכול להסיר אותה.
  4. יש לשים לב כי צד ימין במצב זה הוא מורכב אך ורק מתכונה אחת כי אנחנו בשלב 3 ובשלב 2 פיצלנו את צד ימין לכל התלויות בקבוצה.
4. נקרא לפונקציה `skip_redundant` אשר תדאג לבדוק את התלויות בקבוצת התלויות שקיבלנו עד כה בתהליך, בשאלה: האם תלות בקבוצה ניתנת להסקה מתוך  $F$  לאחר הסרתה מקבוצת התלויות.

פונקציה זו היא הפונקציה העיקרית אשר מסיקה את הכיסוי הקנוני שנקבל מהאלגוריתם.

הבדיקה נעשית ע"י הפעולות הבאות:

- האלגוריתם מסיר את התלות מהקבוצה.
  - האלגוריתם מתבונן על קבוצת התכונות  $\alpha$  של התכונה השמאלית ביחס המתבסס על קבוצת התלויות לאחר ההסרה של התלות בשלב א'.
  - כלומר: אם נסיר את  $X \rightarrow A$ , אז נחשב את  $+X$ .
  - אם  $A$  תהיה מוכלת בקבוצה של  $+X$ , אזי ניתן להסיק אותה ולכן נמחק את היחס, כלומר "נדלג" עליו ולא נשים אותו בקבוצת היחסים הסופית של הכיסוי המינימלי של  $F$ .
  - אם  $A$  לא תהיה מוכלת בקבוצה של  $+X$ , אזי תלות זו איננה מיותרת ולכן היא שייכת לכיסוי הקנוני של קבוצת התלויות  $F$ .
5. לאחר שקיבלנו את הכיסוי הקנוני, נמיין ע"י שימוש בפונקציה `sort` את קבוצת התלויות לפי סדר האלפבית ונשים את קבוצה זו בתוך המשתנה `minimal` שהגדרנו.

ד. `merge chars`

נחזיר את קבוצת התלויות שקיבלנו לצורה יותר "יפה" מאשר צורה קנונית כקבוצה סופית – כיסוי קנוני של  $F$ .

לסיים, דוגמא לעץ הוכחה בפרולוג עבור קבוצת החזקה של הקבוצה [a,b]

