***המחלקה להנדסת תוכנה***

***פרוייקט גמר – תשע"ח***

חישוב מודלים מינימלים באמצעות גרף התלויות

Modular Construction of Minimal Models

**מאת: עדי טיירי**

**מנחה אקדמי: פרופ' רחל בן אליהו זהרי אישור: תאריך:**

**רכז הפרויקטים: ד"ר שפנייר אסף אישור: תאריך:**

עבודה משותפת עם הפרויקט של עמרי מזרחי

*מערכות ניהול הפרויקט:*

|  |  |  |
| --- | --- | --- |
| # | מערכת | מיקום |
| 1 | מאגר קוד | <https://github.com/mazmaz2k/Modular-Construction-of-Minimal-Models.git> |
| 2 | יומן | <https://calendar.google.com/calendar/r/month/2018/1/1?cid=azlodGIxNG5qaHRldGU2bW1ndmk2NTlhdDhAZ3JvdXAuY2FsZW5kYXIuZ29vZ2xlLmNvbQ> |
| 5 | סרטון גרסת אלפא | <https://www.dropbox.com/s/fo1g5yzg1txh01g/%D7%A1%D7%A8%D7%98%D7%95%D7%9F%20%D7%A2%D7%9C%20%D7%94%D7%A4%D7%A8%D7%95%D7%99%D7%99%D7%A7%D7%98-%D7%A2%D7%93%D7%99.mp4?dl=0> |

תקציר

הפרוייקט שנעשה הוא פרוייקט מחקרי ,שנעשה בשיתוף פעולה עם הסטודנט עמרי מזרחי והמנחים שלנו פרופ' רחל בן אליהו זהרי וד"ר יהודה חסין.

אנו נממש את האלגוריתם כפי שמוצע במאמר [1] , מטרת האלגוריתם הוא מציאת מודל מינימלי לאוסף חוקים בצורת CNF ,תוך כדי שיפור זמן הריצה של האלגוריתם.

מציאת המודל המינימלי תעשה על ידי בנית גרף לסט החוקים עפ"י סדר מסוים( הפניה לפרוייקט של עמרי) וזמן הריצה של האלגוריתם תלוי ברכיבי הקשירות הקיימים בגרף.

הבעיה המרכזית העומדת בפני פרויקט זה היא קודם כל מימוש האלגוריתם בצורה טובה למציאת המודל המינימלי, מעבר לכך, זמן הריצה של האלגוריתם המוצע הינו בזמן מעריכי בגודל רכיב הקשירות הגדול ביותר. בהרבה מקרים, עבור סט חוקים, רכיב הקשירות הגדול ביותר הינו משמעותי מאוד ביחס לגרף ,ז"א יכול להיות מצב בו כל הגרף הוא רכיב קשירות אחד גדול ,ולכן זמן הריצה שם הוא מעריכי גבוה ביותר.

פרויקט זה נעשה בשיתוף פעולה עם הפרויקט של עמרי מזרחי, כאשר בפרויקט של עמרי נעסוק בחלק של בניית הגרף, מציאת רכיבי הקשירות שם, וכן פירוק רכיבי קשירות גדולים בגרף.

כיוון שזמן הריצה של האלגוריתם מעריכי בגודל רכיב הקשירות הגדול ביותר לכן ע"י שיתוף פעולה בין שני הפרויקטים יהיה ניתן להקטין את זמן הריצה הכולל של האלגוריתם, ואת זה אנו נבדוק.

ניהול שני הפרויקטים דורשים המון סדר ואחריות, יש חובה לסנכרן בין הפרויקטים השונים, על מנת להגיע למצב ששני האלגוריתמים יעבדו יחדיו.

מילון מונחים , סימנים וקיצורים:

פורמט החוקים יהיה בצורה הבאה:

כך ש .m>0

החלק שלפני החץ נקרא body והחלק שאחריו נקרא head .

מודל – השמה מספקת לסט החוקים.

מודל מינימלי – הוא מודל עבורו מספר ההשמות של ערכי TRUE במשתנים עבור סט של חוקים הוא מינימלי (קיים הסבר מפורט יותר במבוא).

רכיב קשירות- בגרף רכיב קשירות הוא אוסף של קדקודים שמכל קדקוד ניתן להגיע לקדקוד אחר.

סופר גרף- גרף של רכיבי קשירות כל קדקוד יהיה בעצם רכיב קשירות בגרף המקורי.

Source – קדקוד בסופר גרף אשר לא נכנסים אליו קשתות אלה רק יוצאים ממנו קשתות.

# מבוא

בפרויקט זה נתייחס לבעיית הSAT - או בעברית [בעיית הספיקות](https://he.wikipedia.org/wiki/%D7%91%D7%A2%D7%99%D7%99%D7%AA_%D7%94%D7%A1%D7%A4%D7%99%D7%A7%D7%95%D7%AA), בעיה זו הוכחה כ- NP שלמה (קוק-לוין) משמעות זו היא שלא קיימת לבעיה זו אלגוריתם שפותר אותה בזמן שאינו מעריכי.

לבעיית מציאת מודלים מינימלים של בעיות לוגיות (SAT) יש היסטוריה ארוכה. אלגוריתמים חדשים מסוימים לפתרון בעיות SAT כגון :GSAT[SLM92], WALKSAT[SKC94], וגרסאות משופרות של Davis Putnam[DLL62,CA93,LA97] מאפשרים לנו לפתור בעיות SAT גם עבור סט גדול של חוקים.

חישוב מודלים מינימליים הוא נושא מרכזי בבינה מלאכותית (AI), ועומד במרכזם של מערכות רבות כגון logic programming ,, default reasoning planning minimal diagnosis .

במודל זה אנו נתייחס לתיאוריה עבור סט חוקים מהצורה הבאה כאשר m>0 .

כל חוק מהצורה הנ"ל ניתן להמיר לחוק בצורת clause בודד של צורת CNF.

*דוגמא לצורת CNF עבור המשתנים* x1…xn : (x1 v ~x4)^(x3 v x5 v ~x2)^ (xn) כל חלק שנמצא בתוך סוגרים נקרא clause .

מודל מינימלי (הסבר פורמלי)

בהנחה ש m הוא מודל של תיאוריה מסוימת T , נגדיר positive (m) להיות קבוצת המשתנים להם m מציבה True . נאמר ש M היא קבוצת כל המודלים ,אזי *m∈ M* הוא ***מודל מינימלי*** עבור T אם"ם לא קיים מודל *m’∈ M כך ש positive(m’)⊂ positive(m).*

חישוב מודל מינימלי מתחלק לשתי משימות, הראשונה מציאת מודל והשנייה בדיקה האם המודל הוא מינימלי*.*

דוגמא עבור פסוק T מצורת CNF

(-6 v 5) T = (-1 v 2) ^ (-2 v 1) ^ (1 v 3) ^ (-1 v 4 v 5 v 6) ^ (-5 v 6) ^

דוגמא למודל מינימלי: M = {1 , 4} מודל זה הוא מינימלי כי אם נשים ערכי true למשתנים 1 ו- 4 ולכל שאר המשתנים נשים ערכי false אזי T יקבל ערך אמת, ובנוסך לכך, לא קיימת לM תת קבוצה, עבורה T תקבל ערך אמת, אם נשים במשתנים שלה ערכי true ובשאר המשתנים ערכי false .

דוגמא למודל לא מינימלי: M = {1,4,5} מודל זה אינו מינימלי ,מכיוון שקיימת תת קבוצה M’ = {1,4} שאם נשים במשתנים שלה ערכי true ובשאר המשתנים ערכי false נקבל בT ערך אמת.

# תיאור הבעיה

המטרה היא מציאת מודל מינימלי עבור סט של חוקים אך הבעיה הוא זמן ריצה מעריכי גדול מאוד.

אנו מממשים את האלגוריתם שמוצא פתרון בזמן מעריכי בגודל רכיב הקשירות הגדול ביותר , יש מקרים שעבורם רכיב הקשירות הוא בגודל הגרף ולכן זמן הריצה יהיה משמעותית גדול כלומר מעריכי בגודל הקלט.

קיימים פרויקטים המוצאים מודל מינימלי ע"י שימוש באלגוריתמים קיימים הפותרים את בעיית הSAT- , הבעיה המרכזית בפרויקטים אלו שהם מריצים את האלגוריתמים על כל סט החוקים מלכתחילה ולמרות שיפורים קטנים באלגוריתמים בין פרויקט לפרויקט זמן הריצה לא קטן באופן משמעותי.

חשוב מאוד שנשתמש באלגוריתמים יעילים כדי למצוא ולפרק את הרכיב הקשירות ובכך נשפר את זמן הריצה, כיוון שמטרת הפרויקט הינו שיפור זמני ריצה של אלגוריתם למציאת מודל מינימלי.

* *פרויקט זה מסתעף לשני כיוונים בפתרון הבעיה. הראשון הוא בניית החוקים במבנה נתונים דינמי בו ניתן לשנותו עבור השמה מסוימת של משתנה, והשני הוא בניית הגרף מציאת רכיב הקשירות הגדול וכן מציאת מספר מינימלי של קדקודים החשוד לפירוק הרכיב הקשירות בצורה מאוזנת ככל הניתן [בפרוייקט של עמרי]*

בעיות בבניית מבני הנתונים לחוקים:

קיים אלגוריתם ידוע Davis Putnam לפתרון בעיית SAT שבו אנו משתמשים במערכת החוקים ע"מ למצוא מודל מינימלי, אנו נצטרך לממש את אלגוריתם זה.

בנוסף לאלגוריתם נרצה לשלב את הפרוייקט של עמרי בכדי לפרק רכיבי קשירות. יש צורך לקבל מהגרף קדקודים מסוימים ע"י בדיקות שנעשו בפרויקט שלו, שתפקידם הוא לפרק רכיב קשירות , ולכן לאחר קבלת קדקודים שאמורים לפרק את רכיב הקשירות יש צורך להציב בכולם ערכים ( TRUE\FALSE ) . יכול להיות מצב בו עבור השמה מסוימת לא תהיה השמה מספקת עבור כל סט החוקים, ולכן יש לבדוק לאחר ההצבה האם ניתן לקבל השמה מספקת.

יהיה צורך ,כנראה בהמשך הפרויקט, לחפש אלגוריתם המבצע את מציאת המודל המינימלי באופן יותר טוב מבחינת זמן ריצה וזיכרון.

בעיות בבניית מבני הנתונים לגרף:

בפרוייקט של עמרי [Minimal Balanced Node Seperator]

## הבעיה בבדיקות

כל הבדיקות אמורות להתבצע עבור סטים גדולים של חוקים.

אנו נצטרך להשתמש בתוכנה קיימת בשם [WASP](http://alviano.github.io/wasp/) אשר חלק ממטרותיה היא מציאת מודל מינימלי והיא תוכנה קיימת שנמצאת בGitHub - , היא מיישמת טכניקות שיוצגו במקור עבור פתרון SAT בשילוב עם שיטות אופטימיזציה.

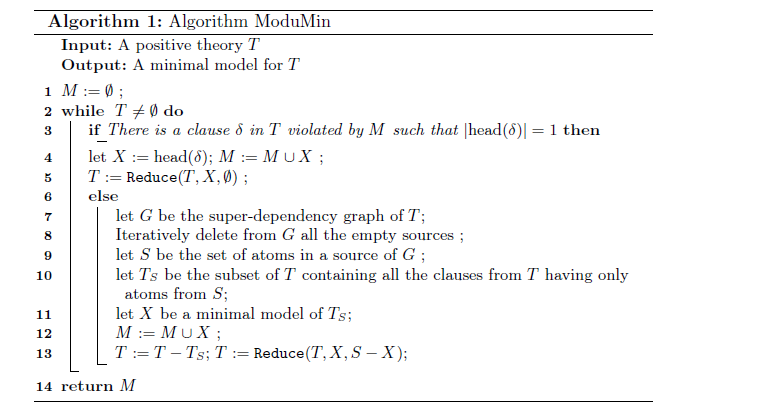
אנו נשווה את תוצאות הבדיקה של האלגוריתם שלנו עם התוצאות של תוכנת WASP .

תוכנה זו נכתבה בשפת ++C ומיושמת בסביבת Linux ולכן יש בעיה בקישוריות בין התוכנה שלנו לתוכנה שלהם.

# תיאור הפתרון

פרויקט זה עוסק בפתרון בעיית SAT , ולכן יש למצוא עבור סט של חוקים האם קיימת השמה מספקת או לא. אנו משתמשים בצורת סט חוקים של CNF ובנוסף לכך עבור כל clause לא קיים מצב בו כל המשתנים הם בשלילה, אם כך כל סט חוקים בצורה זו הוא SAT אך מטרתנו היא **למצוא מודל מינימלי** עבור סט החוקים ,ז"א אנו רוצים לקבל ערך אמת מכל החוקים ע"י כמה שפחות הצבות של true במשתנים הקיימים.

עבור מציאת מודל מינימלי יש להשתמש באלגוריתמים לפתרון בעיית SAT זמן הריצה יהיה מעריכי כיון שבעיה זו הוכחה כבעיה השייכת למחלקה NP שלמה(קוק-לוין) .

**האלגוריתם**: 

**הסבר האלגוריתם**

עבור סט החוקים T , וM המודל המינימלי שבהתחלה ריק.

כל עוד T אינה קבוצה ריקה :

1) נבנה גרף מסט החוקים וכן גרף של רכיבי הקשירות ונחזיר את s (הsource של גרף רכיבי הקשירות.)

2) נמצא את Ts (תת קבוצה של T המכילה את כל החוקים שיש בהם רק משתנים מ s.)

3) אם Ts קבוצה ריקה אז כל המשתנים מs מקבלים ערכי false

4) נמצא את X (מודל מינימלי עבור Ts.)

5) נוסיף למודל המינימלי M את X.

6) נוריד מT את Ts , וכן עבור כל המשתנים שהצבנו בהם ערך false נעדכן את T .

זמן הריצה של אלגוריתם זה הוא מעריכי בגודל רכיב הקשירות הגדול ביותר.

כיוון שיכול לקרות מצב שיש רכיב קשירות ממש גדול אזי Ts יכול להיות גם כן ממש גדול וזמן הריצה יגדל משמעותית.

לכן מטרת שיתוף הפעולה בין שני הפרוייקטים שלנו הוא לקחת את רכיב הקשירות הגדול ביותר ולפרק אותו ע"י השמה של כמה משתנים מתוך הרכיב הקשירות הגדול.

מציאת קדקודים אלו שיכולים לפרק את רכיב הקשירות נעשה ע"י הגרף (הפניה לפרויקט של עמרי[Minimal Balanced Node Seperator])

לאחר מציאת קדקודים אלו יש לבצע השמה של על אותם המשתנים ולעשות reduce , ז"א לצמצם את מבנה הנתונים של החוקים, כיוון שיכול להיות עבור השמה מסוימת של אותם קדקודים סט החוקים עצמו לא יהיה SAT ,לכן עבור השמה של המשתנים יש לבדוק האם קיימת סתירה בהשמה זו או לא ומכיוון שמטרתנו היא מציאת מודל מינימלי לכן נעבור על ההשמות בסדר מסוים כך שיהיו כמה שפחות ערכי false .

## דוגמא לריצת האלגוריתם

זהו גרף התלויות עבור סט החוקים משמאל.

סופר גרף:



ניקח את source {b} , T{b} = {} ,לכן נקבע שערך b הוא false ונעדכן את החוקים והגרף.

M={}

ניקח את source {c} , T{c}= {} , לכן נקבע גם את ערך c להיות false ,נעדכן את החוקים והגרף.

M={}

ניקח את source {a} T{a} = {a} , לכן a יקבל ערך true ,נעדכן את החוקים והגרף, ונוסיף את a למודל המינימלי.

M={a}

ניקח עכשיו את source {d} T{d} = {} ,ולכן ערך d יהיה false , ונעדכן את החוקים והגרף.

M={a}

ואחרון source {e,f} , T{e,f}={{e v f}, {e🡪f}, {f🡪e}} , המודל המינימלי הוא ששניהם true , נעדכן את החוקים והגרף ונוסיף את e ו f למודל המינימלי.

M={a,e,f}

כיוון שסיימנו עם החוקים נחזיר את M כמודל המינימלי.

# תכנית בדיקות

את הבדיקות נוכל לעשות רק על פסוקיות מצורת CNF .

הקובץ שאותו נקבל יכלול בשורה הראשונה את מספר המשתנים שיש בכל הפסוקיות ואחריו כל שורה מייצגת חוק , בסיום כל חוק נרשום את הספרה "0" ע"מ שנדע שהסתיים החוק (בגיט-user manual מופיע הסינטקס המתאים של הקלט).

חשוב שנשמור על סינטקס מתאים של הקלט ,בגלל שאנו נרצה להריץ את אותו קלט על WASP וכך נדע אם הצלחנו לייעל את האלגוריתם למציאת מודל מינימלי.

עבור כל קובץ בדיקות שמכיל סט של פסוקיות אנו נרצה, בנוסף למציאת מודל מינימלי, למצוא את רכיב הקשירות הגדול ביותר ולבדוק עליו מה קורה שמנסים לפרק אותו.

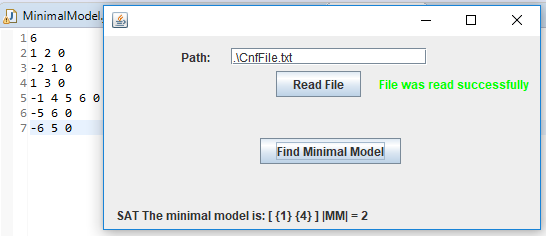
יכולים להיות כמה אפשרויות :

* ברגע שהצלחנו לפרק את רכיב הקשירות הגדול ביותר באופן מאוזן יחסית, אז נצליח לצמצם באופן משמעותי את זמן הריצה המעריכי של האלגוריתם , לזמן ריצה מעריכי אבל קטן משמעותית.
* ברגע שהצלחנו לפרק את רכיב הקשירות אך הפירוק לא היה מאוזן ייתכן שנשאיר רכיב קשירות גדול עדיין , גם אותו ננסה לפרק , וגם הפירוק שלאחריו לא יהיה מאוזן, ככה ברקורסיה (השיטה שמציבה משתנים נעשית ברקורסיה ) . ומצב זה יגרום לנו לבזבז הרבה עבודה על פירוק בלבד ,דבר שיגרום לנו לאבד מזמן הריצה שנרצה לשפר.

דוגמא לפירוק רכיב קשירות:

באיור שלמעלה ניתן לראות גרף שהוא רכיב קשירות, ואם נוציא את הקדקוד האדום אזי נקבל פירוק לגרף בצורה מאוזנת.

## דוגמת הרצה



סט החוקים כמו שניתן לראות בדוגמא הוא בצורת CNF :

(1 v 2) ^ (-2 v 1) ^ (1 v 3) ^ (-1 v 4 v 5 v 6) ^ (-5 v 6) ^ (-6 v 5) T =

נוסחה זו היא SAT וקיים לה מודל מינימלי M {1 , 4} ז"א אם נשים ערך true לערכים 1 ו- 4 ולכל שאר המשתנים נשים ערך false נקבל בT ערך אמת.

בנוסף לכך לא קיימת תת קבוצה ל M שאם נשים במשתנים שלה ערכי true ובשאר המשתנים ערכי false ונקבל השמה מספקת ל T .

# סקירת עבודות דומות בספרות והשוואה

קיימים מספר עבודות שיש להם רמת דמיון לפרויקט שלנו ביניהם

(ראו ביבליוגרפיה ערכים 8,7,24,14,9,3,25,4,1,22,18,34)

WASP:

דוגמה נוספת לעבודה דומה היא התוכנה WASP אשר חלק ממטרותיה היא מציאת מודל מינימלי והיא תוכנה קיימת שנמצאת בGitHub - , היא מיישמת טכניקות שיוצגו במקור עבור פתרון SAT בשילוב עם שיטות אופטימיזציה.

יהיו לנו כמה שימושים תוכנה זאת:

1. לאחר שניצור גרף ונחזיר source (הכוונה לרכיב קשירות בסופר גרף ) , נרצה להציב ערכים במשתנים שהוחזרו בsource . ובכך אולי נצליח לייעל את האלגוריתם WASP כי הוא לא ירוץ על כל סט החוקים וכל המשתנים אלה ירוץ רק על המשתנים שנמצאים ב source .
2. נרצה לעשות בדיקות של האלגוריתם שלנו שמוצא מודל מינימלי באמצעות גרף ובזמן ריצה של רכיב הקשירות בגדול ביותר, לעומת זמן ריצה של אלגוריתם WASP שיפעל ללא כל השיפורים שלנו. כך נדע בעצם אם הצלחנו לשפר את זמני הריצה של האלגוריתם , ואולי נדע על אילו מקרים כן הצלחנו לשפר ועל אילו מקרים לא ומדוע .

Davis Putnam:

דוגמה לעבודה דומה למה שאנו עושים בפרויקט זה ואף אנו משתמשים בו ,הוא האלגוריתם של Davis Putnam , שהרבה משתמשים באלגוריתם זה אף על מנת למצוא מודל מינימלי לסט של חוקים בכך שהוא מתעדף בהצבה של המשתנים ערכי false , אך הוא מתאים לבעיות SAT ששונות מצורת סט החוקים שאנו נשתמש בהם כיוון שאצלנו תמיד יש מודל לכל סט חוקים רק צריך למצוא מודל מינימלי.

מה שלנו יש להציע הוא זמן ריצה יותר מהיר מזמן הריצה של אלגוריתם זה אך במקרים מסוימים במהלך הפרויקט כנראה שנשתמש באלגוריתם זה כדי לראות אם קיים מודל עבור השמה מסוימת.

# סיכום \ מסקנות

בשלב זה של הפרוייקט אנו לוקחים את סט החוקים אשר נמצא בצורתCNF מתוך קובץ, מעבירים אותו למבנה נתונים שהוא שומר בו את החוקים במבנה של body ו head ,מייצרים ממנו גרף ומוצאים מודל מינימלי עבור כל source בגרף ומיד מעדכנים את מבנה הנתונים כך עושים עד שלא נשארו חוקים בכלל, אוסף המודלים המינימלים עבור כל source יוחזר חזרה כמודל המינימלי הכללי של כל סט החוקים שהיה בקובץ.

בפרויקט זה מומש האלגוריתם של Davis Putnam (DLL) ,שמטרתו היא בעצם לתעדף שהשמה של משתנה תהיה false אך עדיין נקבל שסט החוקים הינו satisfiable ובכך מקבלים מודל מינימלי .

ניתן לראות את הבדל זמני הריצה לפי בניית הגרף ז"א כאשר הגרף של סט החוקים קשיר היטב נקבל זמן ריצה ארוך יותר מגרף של סט חוקים בו כל רכיבי הקשירות בו קטנים.

# נספחים

ספרות, תרשימים נוספים, תכנון הפרויקט, טבלת ניהול סיכונים, טבלת דרישות (URD),

## רשימת ספרות \ ביבליוגרפיה

**1**. Modular Construction of Minimal Models by Rachel Ben-Eliyahu-Zohary, FabrizioAngiulli, Fabio Fassett, and Luigi Palopoli .M

F. Angiulli, R. Ben-Eliyahu-Zohary, F. Fassetti, and L. Palopoli. On the tractability

of minimal model computation for some cnf theories. Arti\_cial Intelligence, 2014.

doi: <http://dx.doi.org/10.1016/j.artint.2014.02.003> .

**2**. R. Ben-Eliyahu. A hierarchy of tractable subsets for computing stable models. J.

Artif. Intell. Res. (JAIR), 5:27-52, 1996.

**3**. R. Ben-Eliyahu and R. Dechter. On computing minimal models. Annals of Math-

ematics and Arti\_cial Intelligence, 18:3-27, 1996.

**4**. R. Ben-Eliyahu-Zohary. An incremental algorithm for generating all minimal models.

Arti\_cial Intelligence, 169(1):1-22, 2005.

**5**. R. Ben-Eliyahu-Zohary and L. Palopoli. Reasoning with minimal models: E\_cient

algorithms and applications. Arti\_cial Intelligence, 96(2):421-449, 1997.

**6**. N. Bidoit and C. Froidevaux. Minimalism subsumes default logic and circumscription

in strati\_ed logic programming. In Proceedings of the IEEE symposium on

logic in computer science, pages 89-97, June 1987.

**7**. M. Cadoli. The complexity of model checking for circumscriptive formulae. Inf.

Process. Lett., 44(3):113-118, 1992.

**8**. M. Cadoli. On the complexity of model \_nding for nonmonotonic propositional

logics. In Proceedings of the 4th Italian conference on theoretical computer science,

pages 125-139. World Scienti\_c Publishing Co., October 1992.

**9**. Z. Chen and S. Toda. The complexity of selecting maximal solutions. In Proc. 8th

IEEE Int. Conf. on Structures in Complexity Theory, pages 313-325, 1993.

**10**. M. Davis, G. Logemann, and D. Loveland. A machine program for theoremproving.

Communications of the ACM, 5(7):394-397, 1962.

**11**. J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnoses and systems.

Arti\_cial Intelligence, 56(2-3):197-222, 1992.

**12**. R. Dechter. Constraint processing. Morgan Kaufmann, 2003.

**13**. C. Drescher, M. Gebser, T. Grote, B. Kaufmann, A. Konig, M. Ostrowski, and

T. Schaub. Conict-driven disjunctive answer set solving. KR, 8:422-432, 2008.

**14**. T. Eiter and G. Gottlob. Propositional circumscription and extended closed-world

reasoning are iip2-complete. Theor. Comput. Sci., 114(2):231-245, 1993.

**15**. M. Gebser, B. Kaufmann, and T. Schaub. Advanced conict-driven disjunctive

answer set solving. In IJCAI, 2013.

**16**. M. Gebser, J. Lee, and Y. Lierler. Elementary sets for logic programs. In Proceed-

ings of the 21st National Conference on Arti\_cial Intelligence (AAAI), 2006.

**17**. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive

databases. New Generation Computing, 9:365-385, 1991.

**18**. E. Giunchiglia and M. Maratea. Sat-based planning with minimal-#actions plans

and "soft" goals. In AI\*IA, pages 422-433, 2007.

**19**. T. Janhunen, E. Oikarinen, H. Tompits, and S. Woltran. Modularity aspects of

disjunctive stable models. Journal of Arti\_cial Intelligence Research, pages 813-

857, 2009.

**20**. M. Kalech and G. A. Kaminka. On the design of coordination diagnosis algorithms

for teams of situated agents. Arti\_cial Intelligence, 171(8):491-513, 2007.

**21**. H. A. Kautz, D. Mcallester, and B. Selman. Encoding Plans in Propositional Logic.

In Proceedings of the Fifth International Conference on the Principle of Knowledge

Representation and Reasoning (KR'96), pages 374-384, 1996.

**22**. L. M. Kirousis and P. G. Kolaitis. The complexity of minimal satis\_ability problems.

Inf. Comput., 187(1):20-39, 2003.

**23**. C. Koch, N. Leone, and G. Pfeifer. Enhancing disjunctive logic programming

systems by sat checkers. Arti\_cial Intelligence, 151(1):177-212, 2003.

**24**. P. G. Kolaitis and C. H. Papadimitriou. Some computational aspects of circumscription.

J. ACM, 37(1):1{14, 1990.

**25**. N. Leone, P. Rullo, and F. Scarcello. Disjunctive stable models: Unfounded sets,

\_xpoint semantics, and computation. Inf. Comput., 135(2):69{112, 1997.

**26**. V. Lifschitz. Computing circumscription. In IJCAI-85: Proceedings of the inter-

national joint conference on AI, pages 121-127, 1985.

**27**. V. Lifschitz and H. Turner. Splitting a logic program. In ICLP, volume 94, pages

23-37, 1994.

**28**. J. McCarthy. Circumscription - a form of non-monotonic reasoning. Arti\_cial

Intelligence, 13:27{39, 1980.

**29**. J. McCarthy. Application of circumscription to formalizing common-sense knowledge.

Arti\_cial Intelligence, 28:89{116, 1986.

**30**. R. Reiter. A logic for default reasoning. Arti\_cial Intelligence, 13(1{2):81-132,

1980.

**31**. P. Simons, I. Niemela, and T. Soininen. Extending and implementing the stable

model semantics. Arti\_cial Intelligence, 138(1):181-234, 2002.

**32**. R. T. Stern, M. Kalech, A. Feldman, and G. M. Provan. Exploring the duality in conict-directed model-based diagnosis. In AAAI, 2012.

**33**. *Strong Bridges and Strong Articulation Points of Directed Graphs*, Giuseppe F. Italiano Univ. of Rome “Tor Vergata” . Based on joint work with Donatella Firmani, Luigi Laura, Alessio Orlandi and Federico Santaroni.,

**34**. Davis Putnam, <http://www.jstor.org/stable/1970289?seq=1#page_scan_tab_contents>

***35****. M. Alviano, C. Dodaro, N. Leone, and Francesco Ricca:* **Advances in WASP.** Proceedings of LPNMR (2015). [Download](http://link.springer.com/chapter/10.1007%2F978-3-319-23264-5_5) [Reference](http://alviano.github.io/wasp/wasp2.bib)

***36****. M. Alviano, C. Dodaro, J. Marques-Silva, and F. Ricca:* **Optimal Stable Model Search: Algorithms and Implementation.** Journal of Logic and Computation (In Press 2015). [Download](http://logcom.oxfordjournals.org/content/early/2015/08/31/logcom.exv061.short?rss=1) [Reference](http://alviano.github.io/wasp/jlcweakconstraints.bib)

**37***. M. Alviano, C. Dodaro, and Francesco Ricca:* **Anytime Computation of Cautious Consequences in Answer Set Programming.** Theory and Practice of Logic Programming (2014). [Download](http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=9303142&fileId=S1471068414000325) [Reference](http://alviano.github.io/wasp/cautiousreasoning.bib)

**38***. M. Alviano, C. Dodaro, W. Faber, N. Leone, and Francesco Ricca:* **WASP: A Native ASP Solver Based on Constraint Learning.** Proceedings of LPNMR (2013). [Download](http://link.springer.com/chapter/10.1007%2F978-3-642-40564-8_6) [Reference](http://alviano.github.io/wasp/wasp.bib)

**39***. M. Alviano, C. Dodaro, and Francesco Ricca:* **Comparing Alternative Solutions for Unfounded Set Propagation in ASP.** Proceedings of AI\*IA (2013)

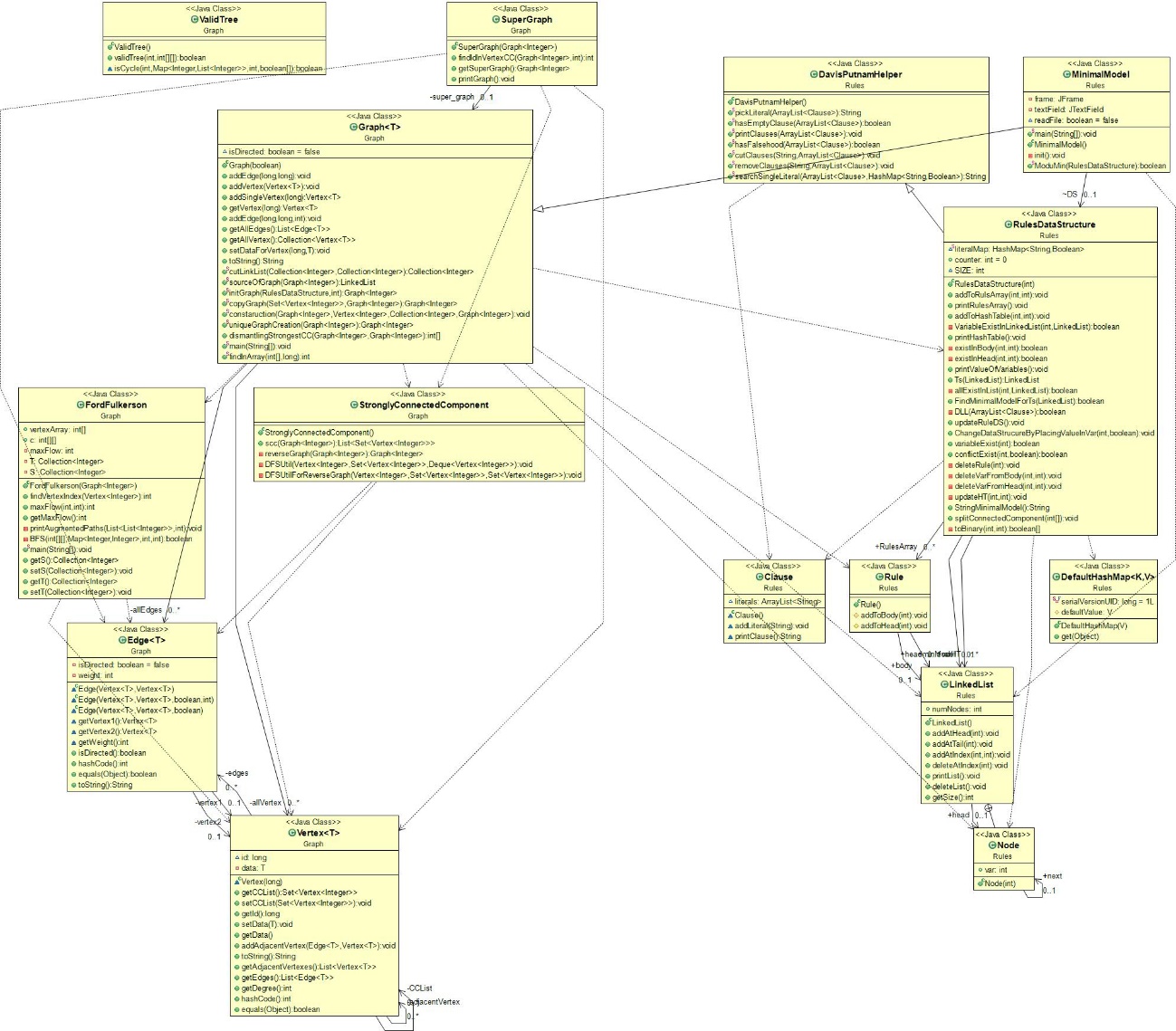
**40**. A Simple Algorithm for Finding All k-Edge-Connected Components :Tianhao Wang,2 Yong Zhang,1,3,\* Francis Y. L. Chin,4,5 Hing-Fung Ting,4 Yung H. Tsin,6 and Sheung-Hung Poon7

## תרשימים וטבלאות

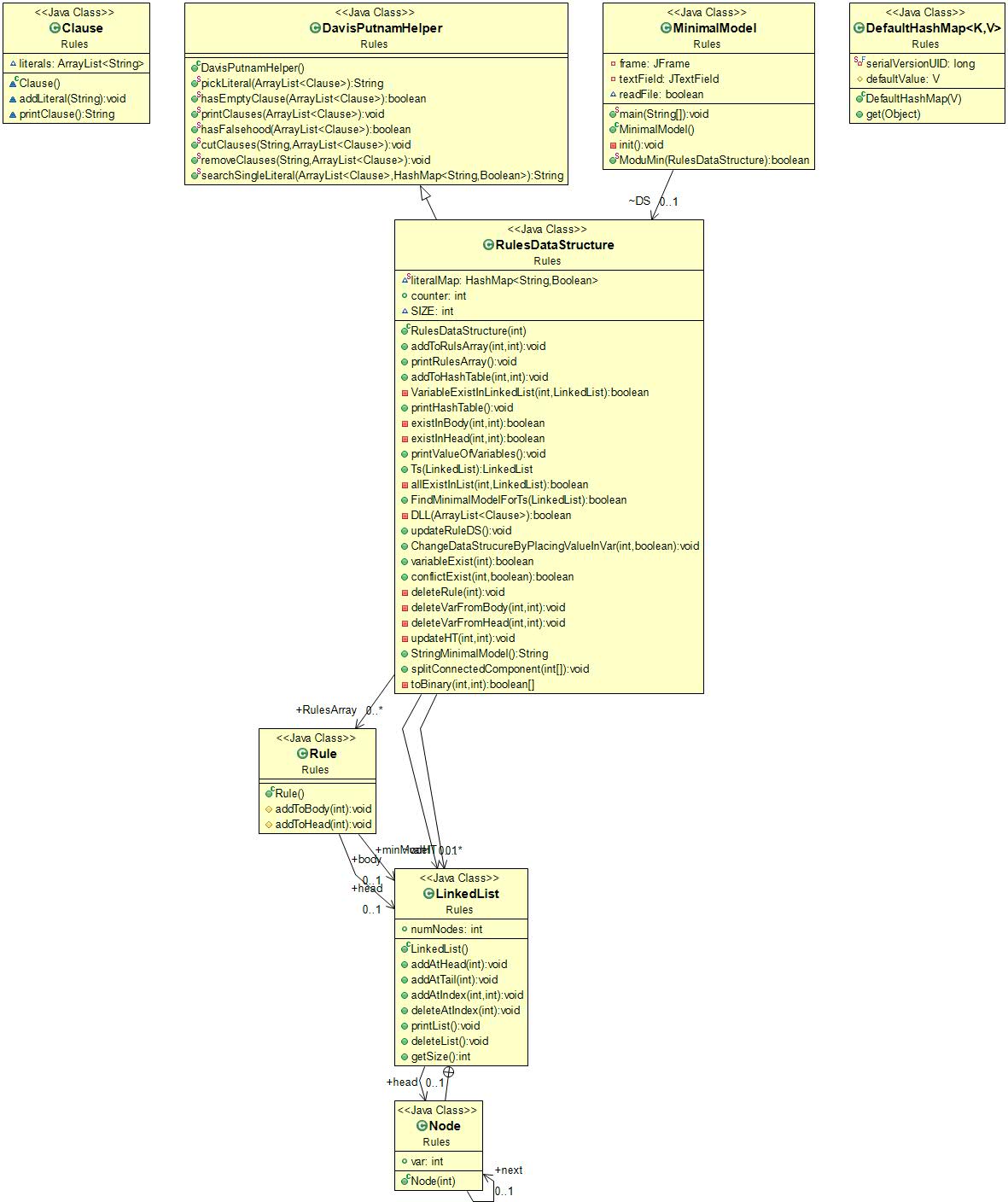
**תרשימי תיכון כגון דיאגרמת רכיבים \ הפצה (UML), דיאגרמת ישויות**

**טבלאות במסד נתונים**

### Class diagram

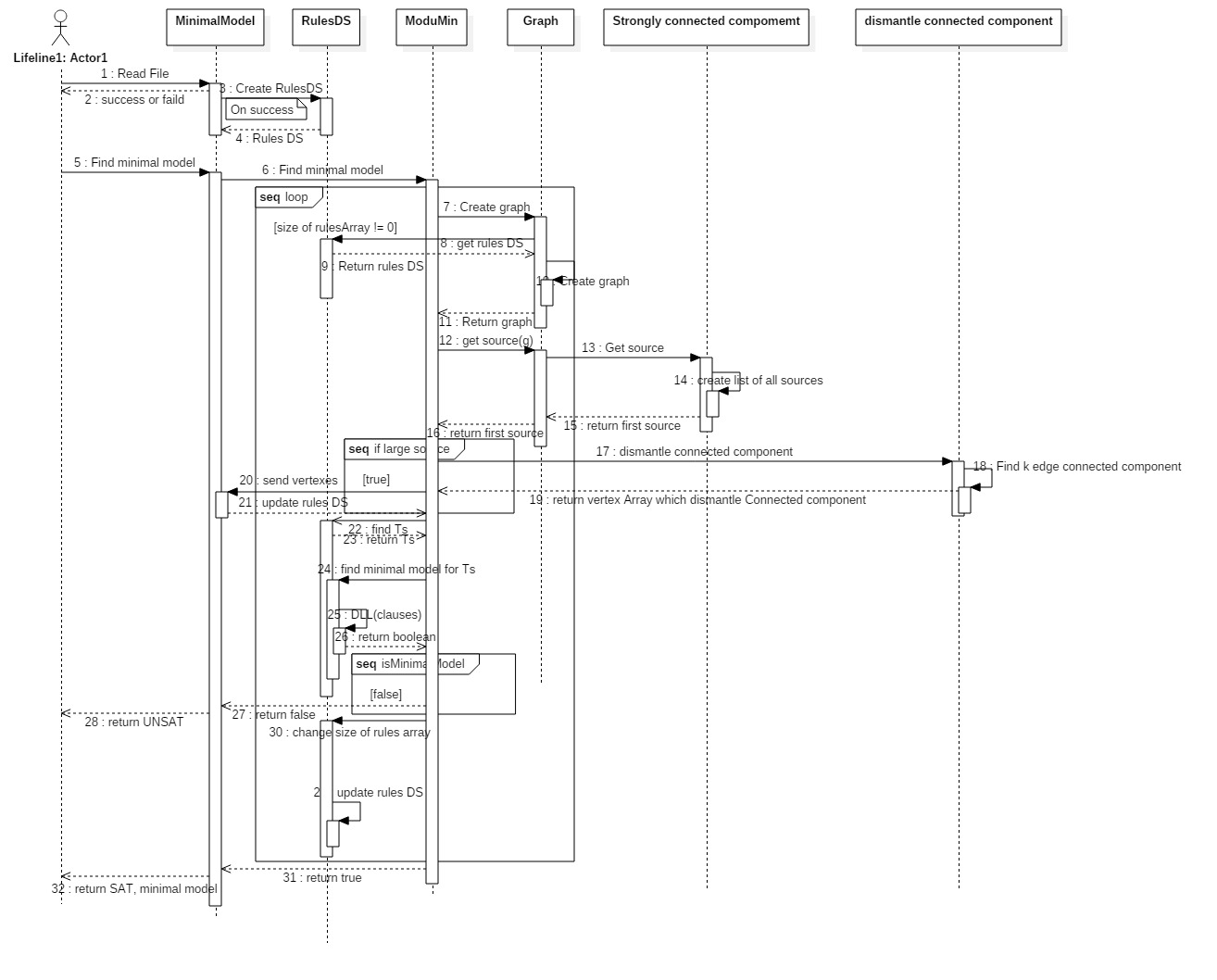


התרשים הנ"ל מייצג את יחסי המחלקות שידועות עבור מחלקת החוקים ומחלקת הגרף.

התרשים הנ"ל מייצג את יחסי המחלקות שידועות עבור מחלקת החוקים.

בנקודת מבט זו ניתן לראות את כל המחלקות שמשתמשות אך ורק בפונקציות של מבנה הנתונים עבור החוקים.

### Sequence diagram



1. בדיאגרמה זאת נתאר את התהליך שבו המשתמש שולח קובץ מוגדר של חוקים, מבנה הנתונים של החוקים יודע לקרוא את הקובץ , להמיר את רשימת כל הפסוקיות בקובץ למבנה נתונים דינאמי .
2. מחלקת מבנה הנתונים של החוקים ישלח את מבנה הנתונים של החוקים למחלקת מבנה הנתונים של הגרף.
3. ניצור גרף שייצג את כל החוקים. כאשר נסיים את בניית הגרף נמצא את רכיב הקשירות הגדול ביותר. כעת יש לנו סופר גרף של רכיבי קשירות והראשון בניהם הוא ה source.
4. אם הsource קטן נציב בכל החוקים של המשתנים שמופיעים בsource ואז נעדכן את סט החוקים .
5. אחרת אם רכיב הקשירות גדול נשלח למחלקת dismantle שמטרתה להחזיר מספר קודקודים שאם נסיר אותם אז נצליח לפרק את רכיב הקשירות. מציאת הקודקודים שבאמצעות שנעבור על כל האפשרויות של ההצבות שלהם נפרק את רכיב הקשירות ואז נעדכן את סט החוקים.
6. אם קיים מודל מינימלי נחזיר אותו אחרת נחזור לסעיף 3 .

## ג. תכנון הפרויקט

|  |  |
| --- | --- |
| 26.7 | פגישת היכרות עם רחל ויהודה– הסבר על תיאורית הפרויקט. |
| 15.8 | פגישה עם רחל ויהודה – בחירת פרויקט |
| פגישות כל שבוע עד תאריך סיום הפיתוח | פגישות קבועות בימי רביעי עם צוות הפיתוח לצורך קידום הפרויקט, פיתוח וחלוקת עבודות. |
| 18.9 | פגישה שלישית עם רחל ויהודה – חלוקה לצוותים  צוות פרויקט על הגרף וצוות פרויקט על החוקים |
| 19.10 | **שלב התנעה** - הגשת טופס התנעה |
| 22.10 | תחילת כתיבת קוד |
| 15.11 | פגישה עם רחל ויהודה עבודה על הצעת הפרויקט |
| 26.11 | **שלב ההצעה** - מסירת נוסח ההצעה, תכנון הפרויקט וניתוח דרישות |
| 1.2 | **שלב האב טיפוס** - מימוש ,מחקר ,סקירת ספרות, ארכיטקטורה ובדיקות-אלפא |
| 20.4 | **שלב הבנייה** - הגשת מסמך תיכון ומימושו |
| [תאריך שיקבע לאחר סיום הפיתוח] | **שלב הבדיקות**- בדיקות של הקוד לבדוק אם הוא משפר זמני ריצה של סטים של חוקים קיימים , השלמת תיעוד/דו"ח טכני |
| 22.6 | **שלב המסירה** – מסירת הפרויקט |

## ד. טבלת סיכונים

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **הסיכון** | **חומרה** | **מענה אפשרי** |
| 1 | עיכוב של תהליכי פיתוח בבניית מבני הנתונים. | 4 | שימוש במבני נתונים קיימים אשר נמצאים באינטרנט. |
| 2 | שילוב אלגוריתם WASP בפרויקט שלנו לא יעבוד כמצופה. | 8 | נעבוד עם אלגוריתם Davis Putnam הפחות יעיל אך שעובד בפרויקט שלנו. |
| 3 | הסנכרון של מחלקת הגרף ומחלקת מבנה הנתונים של החוקים לא יעבוד | 9 | העבודה על בניית המחלקות תעשה בתיאום מוחלט. |
| 4 | עבור בדיקות של כמות גדולה של משתנים לא ימצא מודל מינימלי | 7 | ניצור גרף של סט חוקים זה וננסה לנתח את רכיב הקשירות ומלה לא נמצא מודל כאשר מפרקים את רכיב הקשירות |
| 5 | בחירת סביבת העבודה ושפת התכנות לא מתאימה להרצת סט גדול של חוקים | 3 | מספיק שהתאוריה תעבוד אז יהיה ניתן להעביר את הקוד למגוון של שפות תכנות יותר יעלות |
| 6 |  |  |  |
| 7 | הרצת האלגוריתם שלנו על סביבת Linux לא תעבוד כמצופה. | 2 | לא נוכל להשתמש wasp אך כן נוכל להשתמש באלגוריתמים אחרים שכן הורצו על windows |
| 8 | מטריצת המשקלים שנוצרת מאלגוריתם הזרימה תעמיס מידי על הזיכרון | 6 | נמצא מבנה נתונים יותר יעיל למשקלי הקשתות או שניצור את המטריצה רק פעם אחת ונעבוד איתה |
| 9 | האלגוריתם הרקורסיבי שפועל למציאת k-edge connected component יגדיל במידה גדולה את זמן הריצה שאנו מנסים להוריד | 3 | מספיק שהתאוריה תעבוד אז יהיה ניתן למצוא אלגוריתם יותר יעיל לפתרון |
| 10 | הקדקודים שאם נוציא ,אמורים לפרק את רכיב הקשירות הגדול ביותר , לא יפרקו אותו בצורה מאוזנת . | 2 | נמצא דרך תעזור לנו למצוא קודקודים שבעלי סבירות גדולה יפרקו את רכיב הקשירות בצורה מאוזנת.  עדיין נשפר את זמן הריצה לאחר הפירוק. |

## ה. רשימת\טבלת דרישות

|  |  |  |
| --- | --- | --- |
| מס' דרישה | סוג | תיאור |
| 1 | פלטפורמת מימוש | התוכנה תתבצע בשפת Java בסביבת eclipse |
| 2 | אופן ביצוע | מחלקת החוקים תהיה בסנכרון מלא עם מחלקת הגרף |
| 3 | חלוקת התוכנה | התוכנה תתחלק לשתי כיוונים שונים של עבודה אחד של חוקי התאוריה והשני של גרף המייצג את התאוריה |
| 4 | פונקציונאלי | מבנה הנתונים של הגרף יהיה דינאמי |
| 5 | אופן ביצוע | מבנה הנתונים של הגרף יריץ את האלגוריתם למציאת רכיב הקשירות DFS,וימצא את הגדול ביותר. |
| 6 | אפיון טכני | מבנה הנתונים של הגרף יבנה בהתאם למבנה הנתונים של החוקים  לפי הפורמט של body🡪head |
| 7 | אופן ביצוע | המערכת תתחזק מבנה נתונים השומר עבור כל משתנה באיזה חוק הוא נמצא |
| 8 | פונקציונאלי | מבנה הנתונים של הגרף יפרק את רכיב הקשירות הגדול ביותר . |
| 9 | בדיקות | נריץ בדיקות על מודלים קיימים ע"מ שנראה שקיים שיפור בזמני הריצה. |
| 10 | פונקציונאלי | מחלקת הנתונים של הגרף תחזיר עד שלושה קודקודים בשביל שמערכת החוקים תציב במשתנים הללו. |
| 11 | בדיקות | הרצת האלגוריתם שלנו מול אלגוריתם WASP |
| 12 | פונקציונאלי | שילוב אלגוריתם WASP באלגוריתם שלנו. |
| 13 | אופן ביצוע | שימוש באלגוריתם k-edge connected component לפירוק רכיב הקשירות והחזרת קודקודים שיפרקו את הגרף. |
| 14 | פונקציונאלי | מבנה הנתונים של החוקים יהיה דינמי. |
| 15 | פונקציונאלי | תהיה אפשרות להוריד משתנה או חוק שלם במבנה הנתונים. |