המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ט

אלגוריתם מבוזר לחישוב מודלים מינימליים ומודל מינימום

Distributed Construction of Minimal Models and Minimum Model

מאת

עדי הלאלי

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

רכז הפרויקטים: דר' אסף שפנייר אישור: תאריך:

מערכות ניהול הפרויקט:

#	מערכת	
1	מאגר קוד	https://github.com/adihalali/Distributed-Construction-of-Minimal-Models-and-Minimum-Model
2	יומן	https://github.com/adihalali/Distributed-Construction-of-Minimal-Models-and-Minimum-Model/wiki
3	ניהול פרויקט (אם בשימוש)	https://github.com/adihalali/Distributed-Construction-of-Minimal-Models-and-Minimum-Model/projects/1
4	סרטון גרסת אלפא	https://drive.google.com/file/d/1ZW8TiL4ONPDXhH3n3zkO7a6rSUgqydca/view

תקציר

במסגרת פרויקט גמר במחלקה להנדסת תוכנה במכללת עזריאלי (JCE), הוטלה עלי מטלה לבצע פרויקט במסגרת של 400 שעות.
הפרויקט שאעשה הינו פרויקט מחקרי בהנחיית פרופ' רחל בן אליהו זהרי.

המחקר שייך לתחום הבינה מלאכותית. בעיית המחקר – בהינתן נוסחת SAT חיובית, יש לחשב מודלים מינימליים ככל שניתן ולמצוא את מודל המינימום בה.
מטרת הפרויקט, פיתוח ומימוש אלגוריתם מבוזר לחישוב מודלים מינימליים ומודל מינימום.
חישוב המודלים יבוצע על אוסף חוקים בצורת CNF, עבורם נבנה גרף מכוון המכיל רכיבי קשירות.
מציאת המודלים ימומש באופן מבוזר על הגרף.
בעת פיתוח ומימוש האלגוריתם נדרוש חישוב יעיל למציאת הפתרון.

למציאת מודל מינימלי אעזר באלגוריתם קיים, המוצע במאמר [1] (ראה נספח 7.1).

הבעיה המרכזית העומדת בפני פרויקט זה הינה פיתוח אלגוריתם אמין ומעבר לכך, שזמן הריצה למציאת המודלים יהיה יעיל.

רמת בית הכרם, ת.ד. 3566, ירושלים מיקוד 9103501 Ramat Bet Hakerem, P.O.B 3566, Jerusalem
Tel. +972-2-6588000 | Fax. +972-2-6588001 | Email: info@jce.ac.il | Web: www.jce.ac.il

תוכן עניינים

מילון מונחים, סימנים וקיצורים

פורמט החוקים יהיה בצורה הבאה: $a_1 \wedge \cdots \wedge a_n \rightarrow b_1 \vee \cdots \vee b_m$, כך ש m>0.
החלק שמשמאל החץ נקרא body והחלק שמימנו נקרא head.

- פסוקית – אוסף של ליטרלים (משתנים ושלילת משתנים) המחוברים ביניהם על ידי פעולות "או" ($\wedge$).
- CNF – ביטוי המורכב מאוסף "פסוקיות" המחוברות ביניהן על ידי פעולות "וגם" ($\vee$).
- מודל – השמה מספקת לסט החוקים.
- מודל מינימלי – מודל עבורו מספר ההשמות של ערכי *TRUE* במשתנים עבור סט של חוקים הוא מינימלי.
- מודל מינימום – המודל המינימלי הקטן ביותר באוסף חוקים.
- גרף – ייצוג מופשט של קבוצה של אובייקטים, כאשר כל תת קבוצה של אובייקטים בקבוצה עשויים להיות מקושרים זה לזה.
- גרף מכוון – גרף המורכב מקבוצה של צמתים וקבוצה של קשתות מכוונות, כאשר ישנה משמעות לכיוונה של קשת מכוונת – יוצאת מצומת אחד ונכנסת לצומת אחר.
- רכיב קשירות – קבוצת קדקודים כאשר קיים מסלול בין כל שני קדקודים השייכים לה, וכוללת עם כל קדקוד גם את השכנים שלו.
- גרף רכיבי קשירות – גרף מכוון ובו קיים מסלול מכל צומת שבו אל כל צומת אחר. כלומר, אוסף של קדקודים שמכל קדקוד ניתן להגיע לקדקוד אחר.
- סופר גרף – גרף רכיבי קשירות. כלומר, כל קדקוד מייצג רכיב קשירות בפני עצמו.
- Source – קדקוד בסופר גרף אשר לא נכנסים אליו קשתות, אלא רק יוצאים ממנו קשתות.

1 מבוא

1.1 רקע כללי

חישוב מודלים מינימליים ומודל מינימום עומד במרכזם של כמה מערכות ייצוג ידע, הכוללות circumscription, default logic, minimal diagnosis, planning, multi-agents coordination, logic programs.

מודלים מינימליים, מהווה נושא מרכזי במספר מחקרים בתחום הבינה מלכותית. כמו כן, מציאת המודלים בכלל הינה משימה מרכזית בתחום זה (AI).

פרויקט זה מתייחס לבעיית ה-SAT, או בעברית "בעיית הספיקות" – בעיית הכרעה שהוכחה כ-NP שלמה (קוק-לוין). כלומר, לבעיה זו לא קיים אלגוריתם הפותר אותה בזמן ריצה שאינו מעריכי. מתברר כי משימה זו הינה קשה גם במידה והתיאוריה הנתונה נחשבת לפשוטה.

בפרויקט זה, נתמקד בבעיות SAT חיוביות. כלומר, כל פסוקית מכילה משתנה חיובי אחד לכל הפחות.

לבעיית מציאת מודלים מינימליים של בעיות לוגיות (SAT) קיימת היסטוריה ארוכה. קיימים מספר אלגוריתמים חדשים לפתרון בעיית SAT, כגון: GSAT(SLM92), WALKSAT(SKC94), וגרסאות משופרות של Davis Putnam algorithm [DLL62, CA93, LA97], המאפשרים לנו לפתור בעיות SAT גם עבור סט גדול של חוקים.

1.2 מהי תיאוריה?

במודל זה אנו נתייחס לתיאוריה עבור סט חוקים מהצורה הבאה $a_1 \wedge \cdots \wedge a_n \rightarrow b_1 \vee \cdots \vee b_m$, כאשר $m > 0$.

כל חוק מהצורה הנזכרת לעיל, ניתן להמיר לחוק המוצג כפסוקית בודדת בצורת CNF.

דוגמא לצורת CNF עבור המשתנים $x_1, \dots, x_n$: $(x_1 \vee \neg x_4) \wedge (x_3 \vee x_5 \vee \neg x_2) \wedge (x_n)$. כל ביטוי תחום בסגורים מהווה פסוקית.

1.3 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 $m \in M$ הוא מודל מינימלי עבור *T* אם ורק אם לא קיים מודל $m' \in M$ כך ש $positive(m') \subset positive(m)$.

1.3.1 דוגמא למודל מינימלי

פסוק T בצורת CNF: $\mathbb{T} = (a \vee b) \wedge (\neg b \vee a) \wedge (a \vee c) \wedge (\neg a \vee d \vee e \vee f) \wedge (\neg e \vee f) \wedge (\neg f \vee e)$

המודל המינימלי בT, $M = \{a, d\}$.

הסבר: נציב ערכי *TRUE* (T) במשתנים a, d, ולשאר המשתנים נציב ערכי *FALSE* (T).

$$\mathbb{T} = (T \vee F) \wedge (\neg F \vee T) \wedge (T \vee F) \wedge (\neg T \vee T \vee F \vee F) \wedge (\neg F \vee F) \wedge (\neg F \vee F)$$
$$\mathbb{T} = (T \vee F) \wedge (T \vee T) \wedge (T \vee F) \wedge (F \vee T \vee F \vee F) \wedge (T \vee F) \wedge (T \vee F)$$
$$\mathbb{T} = T \wedge T \wedge T \wedge T \wedge T \wedge T = T$$

קיבלנו מודל מינימלי עבורו $\mathbb{T} = T$. כמו כן, לא קיימת תת קבוצה של M, כך שT תהיה טאוטולוגיה.
כלומר, זהו אכן מודל מינימלי בT.

1.3.2 דוגמא למודל שאינו מינימלי

פסוק T בצורת CNF: $\mathbb{T} = (a \vee b) \wedge (\neg b \vee a) \wedge (a \vee c) \wedge (\neg a \vee d \vee e \vee f) \wedge (\neg e \vee f) \wedge (\neg f \vee e)$
מודל שאינו מינימלי: $M = \{a, d, e\}$.
הסבר: קיימת תת קבוצה $M' = \{a, d\}$, כך שT טאוטולוגיה (ראה דוגמא 1.3.1).

1.4 מודל מינימום (הסבר)

בהנחה שבתיאוריה מסוימת T, קיימים m מודלים מינימליים, אזי מודל המינימום עבור T, יהיה המודל המינימלי שבו מספר המשתנים המקבלים ערך *TRUE* הוא הקטן ביותר.

1.4.1 דוגמא למודל מינימום

פסוק T בצורת CNF: $\mathbb{T} = (a \vee b) \wedge (\neg b \vee a) \wedge (a \vee c) \wedge (\neg a \vee d \vee e \vee f) \wedge (\neg e \vee f) \wedge (\neg f \vee e)$
המודלים המינימליים בT, $M_1 = \{a, d\},\ M_2 = \{a, e, f\}$.
מודל מינימום בT, $M_1 = \{a, d\}$.
הסבר:

- עבור $\mathbb{T}(M_1)$ נציב ערכי *TRUE* (T) במשתנים a, d, ולשאר המשתנים נציב ערכי *FALSE* (T).

$$\mathbb{T}(M_1) = (T \vee F) \wedge (\neg F \vee T) \wedge (T \vee F) \wedge (\neg T \vee T \vee F \vee F) \wedge (\neg F \vee F) \wedge (\neg F \vee F)$$
$$\mathbb{T}(M_1) = (T \vee F) \wedge (T \vee T) \wedge (T \vee F) \wedge (F \vee T \vee F \vee F) \wedge (T \vee F) \wedge (T \vee F)$$
$$\mathbb{T}(M_1) = T \wedge T \wedge T \wedge T \wedge T \wedge T = T$$

- עבור $\mathbb{T}(M_2)$ נציב ערכי *TRUE* (T) במשתנים a, e, f, ולשאר המשתנים נציב ערכי *FALSE* (T).

$$\mathbb{T}(M_2) = (T \vee F) \wedge (\neg F \vee T) \wedge (T \vee F) \wedge (\neg T \vee F \vee T \vee T) \wedge (\neg T \vee T) \wedge (\neg T \vee T)$$
$$\mathbb{T}(M_2) = (T \vee F) \wedge (T \vee T) \wedge (T \vee F) \wedge (F \vee F \vee T \vee T) \wedge (F \vee T) \wedge (F \vee T)$$
$$\mathbb{T}(M_2) = T \wedge T \wedge T \wedge T \wedge T \wedge T = T$$

קיבלנו מודלים מינימליים עבורו $\mathbb{T} = T$. כמו כן, לא קיימת תת קבוצה של M_1, M_2, כך שT תהיה טאוטולוגיה.
כלומר, אכן מודלים מינימליים בT.
$|M_1| = 2 < |M_2| = 3$. כלומר, $M_1 = \{a, d\}$ מודל מינימום של תיאוריה T.

1.4.2 דוגמא למודל מינימלי שאינו מינימום

מתבסס על דוגמא 1.4.1.
פסוק T בצורת CNF: $\mathbb{T} = (a \vee b) \wedge (\neg b \vee a) \wedge (a \vee c) \wedge (\neg a \vee d \vee e \vee f) \wedge (\neg e \vee f) \wedge (\neg f \vee e)$
המודלים המינימליים בT, $M_1 = \{a, d\},\ M_2 = \{a, e, f\}$.
מודל מינימלי בT, $M_2 = \{a, e, f\}$, אינו מודל מינימום בT, מכיוון $|M_2| = 3 \ngtr |M_1| = 2$, שהרי מודל מינימום הוא מודל מינימלי בעל המספר הקטן ביותר של משתנים המקבלים ערכי *TRUE*.

Ramat Bet Hakerem, P.O.B 3566, Jerusalem 9103501 רמת בית הכרם, ת.ד. 3566, ירושלים מיקוד
Tel. +972-2-6588000 | Fax. +972-2-6588001 | Email: info@jce.ac.il | Web: www.jce.ac.il

1.5 מסגרת הפרויקט

הבעיה של חישוב מודל מינימלי או מודל מינימום נמצאת בלב מערכות לייצוג ידע בתחום הבינה המלאכותית. אם משתמשים בתוכניות לוגיות לייצוג ידע, אזי מודל מינימלי הוא הכרחי לחישוב מודלים יציבים של תכניות אלו. אם משתמשים בגישה לוגית לדיאגנוסטיקה של מערכות, אז יש צורך בחישוב מודל מינימום.

הבעיה של חישוב מודל מינימלי ומודל מינימום היא בעיה NP קשה ולכן ככל הנראה אין בנמצא אלגוריתם פולינומיאלי לפתרון הבעיה. מטרת הפרויקט היא לפתח ולממש אלגוריתמים לחישוב המודלים הרצויים אשר יעבדו באופן יעיל עבור תתי קבוצות של תיאוריות לוגיות בעלות מאפיינים גרפיים מיוחדים.

פרטים נוספים על החשיבות של מודל מינימום ומודל מינימלי ניתן למצוא בנספח 7.1 [1].

2 תיאור הבעיה

2.1 בעיית הספיקות

בעיית הספיקות בתחשיב הפסוקים (SAT) הוא שמה של בעיית הכרעה. בעיה זו מוכחת כבעיית NP שלמה.

בתחשיב פסוקים, נוסחה תקרא ספיקה אם קיימת השמה של ערכי אמת עבורה הנוסחה אמיתית.

למשל, הנוסחה $A \vee B$ תקרא ספיקה – ניתן לקבוע את ערך A ל-"אמת" ולכן ערך האמת שיתקבל עבור הנוסחה כולה יהיה "אמת" גם הוא. לעומת זאת, הנוסחה $A \wedge (\neg A)$ אינה ספיקה – לא קימת השמה עבורה ערך A וגם ערך $\neg A$ יהיו יחד בעלי ערך "אמת".

באופן כללי, תיאור בעיית SAT , מתוארת באופן הבא:

בהינתן נוסחה בתחשיב הפסוקים המורכבת מקשרים: "¬" "∧" "∨" , האם קיימת השמת ערכי אמת למשתנים עבורה הנוסחה תהיה טאוטולוגיה?

2.2 דרישות ואפיון הבעיה

2.2.1 אפיון הבעיה

פרויקט זה עוסק בפתרון בעיית SAT. עבור סט חוקים, נרצה לדעת האם קיימת השמה מספקת. כמו כן, מטרת הפרויקט היא מציאת מודל מינימלי ומודל מינימום בתיאוריה. כלומר, אנו דורשים השמה מספקת מינימלית ועבורה נוכל להסיק מודל מינימום.

למציאת מודל מינימלי, יש להשתמש באלגוריתמים לפתרון בעיית SAT. בעיה זו הוכחה כבעיית NP שלמה (קוק-לוין) ועל כן זמן הריצה הינו מעריכי.

כדי למצוא את מודל המינימום בתיאוריה, תחילה יש לחשב עבורה מודלים מינימליים ככול שניתן.

כלומר, לא תמיד ניתן למצוא את כל המודלים המינימליים עבור התיאוריה הנתונה ומכך נובע גם כן, כי לא תמיד נוכל למצוא מודל מינימום.

2.2.2 דרישות עיקריות מהמערכת

- המערכת תאפשר למשתמש להזין את התיאוריה.
- פתרון הבעיה של חישוב המודלים.
- אלגוריתם מבוזר למציאת כל המודלים המינימליים בתיאוריה ומציאת מודל מינימום.
- המערכת תחשב כחלק מתוצאות האלגוריתם את זמן הריצה ואת ניצול הזיכרון.

2.2.3 דרישות מהפרויקט

ראה להלן נספח, 7.5.

2.3 זמני ריצה

קיימים פרויקטים לחישוב מודל מינימלי ע"י שימוש באלגוריתמים קיימים הפותרים את בעיית ה-SAT. הבעיה המרכזית בפרויקטים אלו, שרעיון האלגוריתמים מתבסס על ריצה שמתחילתה מבצעים מעבר על כל סט החוקים. ישנם שיפורים קטנים בזמני הריצה עבור אלגוריתמים שונים ולמרות זאת, בין פרויקט לפרויקט זמן הריצה אינו קטן באופן משמעותי.

2.4 המצב כיום

2.4.1 מודל מינימלי

קיימים מספר אלגוריתמים שונים למציאת מודלים מינימליים ואף מומשו. ראה: נספח 7.1, סקירת עבודות דומות בספרות והשוואה.
פורסם מאמר של פרופ׳ רחל בן-אליהו זהרי למציאת מודל מינימלי ראה: נספח: 7.1-[1]. סטודנט בוגר מהמכללה, מימש את האלגוריתם כחלק מפרויקט הגמר שלו. תוצאות המחקר לא הניבו תוצאות שיפור משמעותיות בזמני הריצה.

2.4.2 מודל מינימום

קיים אלגוריתם למציאת מודל מינימום בתיאוריה, ראה נספח 7.1-[3]. האלגוריתם המתואר במאמר הינו אלגוריתם נאיבי, משמע אלגוריתם שאינו יעיל מבחינת זמן ריצה. כמו כן, עד היום לא פותחה תכנית המממשת את האלגוריתם הנזכר לעיל.

2.5 הבעיה מבחינת הנדסת תוכנה

אנו מעוניינים שהמערכת תעבוד בצורה מבוזרת. עבור תיאוריה נתונה, חישוב כל המודלים המינימליים יבוצע בחלקים ובאופן מקבילי לכדי מציאת מודל מינימום. מאחר והבעיה הוגדרה כבעיה NP קשה, ייתכן ונקבל שתיאוריה מורכבת תהיה פשוטה יותר לחישוב, לעומת תיאוריה פשוטה שתהיה מורכבת יותר לחישוב. כדי לצמצם עלויות חישוב, יש לחשוב היטב על בניית האלגוריתם ביעילות מרבית ושימוש נכון ומתאים של מבני נתונים.
דוגמאות להחלטות שהתקבלו בעקבות בעיות עליהם האלגוריתם נדרש לתת מענה:

- מציאת מודלים מינימליים בתיאוריה יבוצע בצורה מבוזרת.
- הגרף הנבנה מהתיאוריה, לא ישתנה תוך כדי ריצת האלגוריתם.
- המערכת תעבוד על מספר חישובים בו זמנית
  - במטרה לזרז את תהליך החישוב, ניתן לממש את האלגוריתם באופן מקבילי וזאת כדי לא לבצע חישוב יחיד והקטנת זמני המתנה.
  - יש צורך להגביל את העבודה המקבילית, כדי למנוע עבודה מיותרת.
- המערכת תעבוד בצורה א-סינכרונית:
  - מציאת המודלים הינה בעיה NP קשה שזמן הריצה שלה עלול להיות רב.
  - במהלך ריצת האלגוריתם, לא כל החישובים הנעשים תלויים/דורשים בחישובים קודמים.

יש לחקור את צורת הפיתוח לנושאים הבאים:

1. עבודה ב- Multi-threading
2. עבודה א-סינכרונית.
3. שימוש בפונקציות גיבוב על מנת ליעל את השימוש במבני נתונים ובכך לשפר זמני ריצה.

3 תיאור הפתרון

3.1 מהי המערכת

המערכת מורכבת ממשק המפעיל אלגוריתם מבוזר לחישוב מודלים מינימליים ומודל מינימום אפשריים בתיאוריה המורכבת מסט פסוקים. תוצאות המערכת יהיו מודל מינימום (אם קיים), זמני ריצה וניצול זיכרון.
כדי למצוא מודל מינימום בתיאוריה, ראשית יש למצוא את כל המודלים המינימליים. המודל המינימלי הקטן ביותר (אם קיים) יהיה גם כן מודל המינימום.

3.1.1 יעדים עיקריים

- תשתית מרובת תהליכונים לצרכי חישוב הכוללת העברת מסרים.
- בניית מבנה נתונים תוך דגש על יעילות פונקציונאלית.
- בניית אלגוריתם המחשב את המודלים המינימליים כדי למצוא את המודל מינימום.
- דגש על יעילות האלגוריתם.

3.1.2 ארכיטקטורת המערכת

ראה להלן נספח, 7.2.1.

3.2 תהליכים ונתוני המערכת

עבור תיאוריה המורכבת מסט חוקים המערכת תבצע מספר עבודות:

- מציאת מודלים מינימליים באמצעות חישוב מבוזר
- מציאת מודל מינימום
- מדידה והשוואה של זמני ריצה

3.3 תיאור הפתרון המוצע

בניית מערכת מבוזרת, אשר תפעיל אלגוריתם למציאת כל המודלים המינימליים הקיימים בתיאורה T, ובאמצעות המודלים המינימליים תחשב את המודל מינימום בT.

חישוב מודלים מתחלק למספר משימות:

- חישוב מודלים מינימליים:
  - מציאת מודל מינימלי
  - בדיקה האם המודל שנמצא אכן מינימלי
- מציאת מודל מינימום: בהנחה שמצאנו מודלים מינימליים, ניתן לחשב את מודל המינימום בתיאוריה

במאמר שפורסם על ידי פרופ' רחל בן-אליהו-זהרי ושותפיה (ראה נספח 7.1 - [1]) , מוצע פתרון למציאת מודל מינימלי. הרעיון: חילוק משימה למספר תתי-משימות המחושבות בנפרד ובאפון מקבילי. ההצעה: המרת התיאוריה לגרף רכיבי קשירות (סופר גרף SG) ובהתאם למשתנים השייכים ל-source ב-SG, יש להציב ערכים בוליאניים בחוקים המכילים אותם. למעשה, עבור אלגוריתם [1], זמן הריצה תלוי בגודל רכיב הקשירות הגדול ביותר.

דוגמת הרצה – ראה נספח 7.6.

3.3.1 חישוב מבוזר למציאת מודלים מינימליים ומודל מינימום

עבור תיאוריה T:

- אתחול: האלגוריתם מתבסס על גרף מכוון, בעל רכיבי קשירות וללא מעגלים עבור תיאוריה. כל אטום בסט הפסוקיות של התיאוריה יהווה קדקוד וכל חוק יתואר על ידי קשתות מכוונות. עבור כל מעגל בגרף ניצור רכיב קשירות, כדי לקבל גרף מכוון, חסר מעגלים ובעל רכיבי קשירות – סופר גרף.
- חישוב: את המודלים המינימליים נחשב באמצעות שילוב מודלים מינימליים חלקיים. עבור כל קדקוד (רכיב קשירות) בסופר גרף, נאתחל את התאוריה לפי שילוב מודלים חלקיים שהתקבלו מהבנים ובהתאם נחשב מודלים חלקיים שגם כן ישולבו וישלחו לאבא להמשך חישוב. תוצאות האלגוריתם יהוו את המודלים המינימליים בתיאוריה.
  האלגוריתם הינו אלגוריתם חישובי מבוזר. החישוב המבוזר ימומש באמצעות תהליכונים. לכל קדקוד בגרף ניצור תהליכון חישובי.

- שלב ראשוני (עבור קדקודים שאינם source): המתנה לקבלת מודלים חלקיים מהבנים ושילוב למודל M.
- עבור כל מודל משולב M:
  - אתחל את תיאוריה T.
  - חשב את T_s – תת קבוצה המורכבת מכל החוקים המכילים את המשתנים מקדקוד s.
  - חישוב מודלים מינימליים לT_s.
  - שילוב המודלים שהתקבלו עם מודל M.
  - שליחת המודל המשולב לאבא.

- מציאת מודל מינימום: עבור כל המודלים המינימליים שהתקבלו, המודל עבורו מספר המשתנים המקבילים ערכי True הוא הקטן ביותר.

3.3.2 אלגוריתם לחישוב מודל מינימלי חלקי

עבור כל קדקוד בגרף נוצר תהליכון לחישוב מודל מינימלי חלקי. להלן פסאודו-קוד המציג את תהליך החישוב.

$\boldsymbol{Algorithm}: Sub - Minimal - Models$

$\boldsymbol{Input}: current\ vertex$
$\quad parents\ vertexes: p_1, \dots, p_m$
$\quad children\ vertexes: c_1, \dots, c_n$
$\boldsymbol{Output}: A\ sub\ minimal\ model\ of\ current\ vertex\ of\ T$

$\quad Let\ Cm_1, \dots, Cm_n\ be\ the\ sub - models\ received\ from\ children;$

$\quad while\ receive\ sub - models\ from\ children, do:$
$\qquad M := Combine(Cm_1, \dots, Cm_n);$
$\qquad T := Reduce(T, M);$
$\qquad M' := Construct\ minimal\ models\ of\ T;$
$\qquad send\ to\ p_1, \dots, p_i: Combine(M, M');$

3.4 תיאור הכלים המשמשים לפתרון

- בניית גרף רכיבי קשירות עבור סט חוקים:
בפרויקט גמר משנה שעברה, מומש אלגוריתם ליצירת גרף. על כן, אסתמך על מימוש זה בכלי עבור הפרויקט.
- תקשורת בין תהליכונים: (הודעות-מודלים חלקיים)
  - מבנה שרת-לקוח מרובה תהליכונים:
    - שרת לניהול העברת הודעות בין תהליכונים.
    - כל קדקוד בגרף מהווה לקוח, כאשר כל לקוח הוא תהליכון.

- Socket: העברת הודעות בין תהליכונים. עבור כל לקוח נפתח socket עם השרת. את ניתוב ההודעות בין תהליכונים מבצע השרת לשליטה ותיאום ההודעות בצורה אמינה.
- טבלת גיבוב: לשימוש בפונקציות יעילות לפעולות הכנסה/הוצאה/מחיקה/חיפוש של ההודעות המתקבלות אצל הלקוח.

4 תכנית בדיקות

בפרויקט זה קיימת דרישה של ייעול מקסימלי. נרצה לבחון האם לאלגוריתם מבוזר ישנה השפעה על זמני הריצה בחישוב מודלים. כדי להשיג מטרה זו, אבצע בדיקות יחידה עבור כל שלב באלגוריתם.
כדי להגיד את רמת האמינות של התוכנה, נבצע בדיקות משני סוגים: בדיקות עבור תוצאות התכנית ובדיקות לצורך השוואת זמני ריצה. במידה ונמצא שיפור משמעותי נעמוד במבחן התוצאה.

4.1 תשתית לחישוב מבוזר:

בניית התשתית נעשית בהתאם למבנה גרף רכיבי הקשירות. עבור כל קדקוד בגרף נחשב מודלים מינימליים באופן מקבילי. כלומר, מספר התהליכונים יהיה כמספר הקדקודים. בהתאם לקשתות הגרף ניישם רשת תקשורת בין תהליכונים לצרכי העברת חישובי המודלים.
תשתית החישוב הינה בסיס מרכזי במימוש האלגוריתם. לכן, יש להגדיר את רמת אמינותה באמצעות בדיקות יחידה:

- שרת-לקוח מרובה תהליכונים בהתאם לגרף
- העברת הודעות בין תהליכונים לפי אבות ובנים בגרף

4.2 טבלת גיבוב

לפרויקט זה קיימת דרישה לייעול מקסימלי של זמני ריצה. כלומר, מימוש טבלת גיבוב כמבנה נתונים יכול להיות חיוני מאוד לתוצאות זמני הריצה. אבצע בדיקות לצורך אימות זמן הריצה עבור כל פונקציה של טבלת הגיבוב:

- פונקציית חיפוש – זמני ריצה: במקרה הממוצע $O(1)$ ובמקרה הגרוע $O(n)$.
- פונקציית הכנסה – זמני ריצה: במקרה הממוצע $O(1)$ ובמקרה הגרוע $O(n)$.
- פונקציית מחיקה – בהתאם לסיבוכיות של פונקציית החיפוש.

4.3 מציאת מודלים מינימליים

החישוב מתחלק לתתי משימות. כלומר, שילוב של מודלים חלקיים בכל חישוב. תחילה אבצע בדיקות יחידה לצורך אימות המודלים ולאחר מכן על מנת לבחון האם קיים שיפור בזמני הריצה.

4.4 אלגוריתם נאיבי

מימוש האלגוריתם הנאיבי המתואר במאמר [3] (ראה נספח 7.1). באמצעות המימוש נוכל לבצע השוואה של זמני הריצה ולהגיע למסקנות חדשות בנוגע לחישוב מבוזר של הבעיה.

5 סקירת עבודות דומות בספרות והשוואה

5.1 סקירה

ישנם מספר עבודות דומות לפרויקט זה המתקשרות לחישוב מודלים מינימליים.

5.1.1 Modular Construction of Minimal Models

פרויקט גמר בשנת 2018, מימוש אלגוריתם לחישוב מודל מינימלי המתואר במאמר [1] (ראה נספח 7.1).

בשנה שעברה, סטודנט במכללה מימש את האלגוריתם למציאת מודל מינימלי ובדיקתו בהתאם למאמר המפורסם [1]. הפרויקט הנוכחי מתבסס על המאמר גם הוא. על כן, יהיו כמה חישובים ממומשים בהם אעזר במהלך פרויקט זה:

- קבלת חט חוקים בצורת CNF, הכנסה למבנה נתונים ויצירת גרף.
- מציאת כל רכיבי הקשירות בגרף ויצירת סופר גרף.

5.1.2 Davis Putnum

אלגוריתם Davis Putnum הינו אלגוריתם למציאת מודל מינימלי באמצעות תעדוף הצבת ערכי FALSE במשתנים. אלגוריתם זה מתאים לבעיות SAT, אך צורת סט החוקים בבעיות אליהן מתאים שונה מהצורה בפרויקט זה. בפרויקט זה קיים מודל לכל סט חוקים, נדרש רק מציאת המודל המינימלי. במהלך פיתוח הפרויקט, ייתכן ואעזר באלגוריתם זה על מנת לבדוק האם קיים מודל עבור השמה מסוימת.

5.1.3 WASP

תוכנה קיימת הנמצאת בGitHub, אשר אחת ממטרותיה היא מציאת מודל מינימלי.

שימושים אפשריים בחלק מפיתוח הפרויקט:

- חישוב מודלים מינימליים חלקיים. אולי עבור סט חוקים קטן האלגוריתם יבצע חישובים יעילים יותר.
- בדיקות והשוואה של האלגוריתם מול המימוש בפרויקט זה.

6 סיכום ומסקנות

ניתן להגיד כי כל מודל מינימום הינו מודל מינימלי, אך לא כל מודל מינימלי הוא מודל מינימום.

6.1 סיכום שלב אלפא

- כתיבת האלגוריתם בפסאודו קוד.
- יצירת תקשורת בין תהליכונים.
- מימוש טבלת גיבוב לפי צרכי הפרויקט

6.2 מסקנות שלב אלפא

- שימוש בתהליכונים היה עדיף על פני תהליכים כדי ליעל זמני ריצה ואף זמני חישוב.
- בתחילת הפרויקט חשבנו לממש בפייתון, אך פייתון הינה שפה single Threaded ועל כן אינה מומלצת לפיתוח של ריבוי תהליכונים.

7 נספחים

7.1 רשימת ספרות \ ביבליוגרפיה

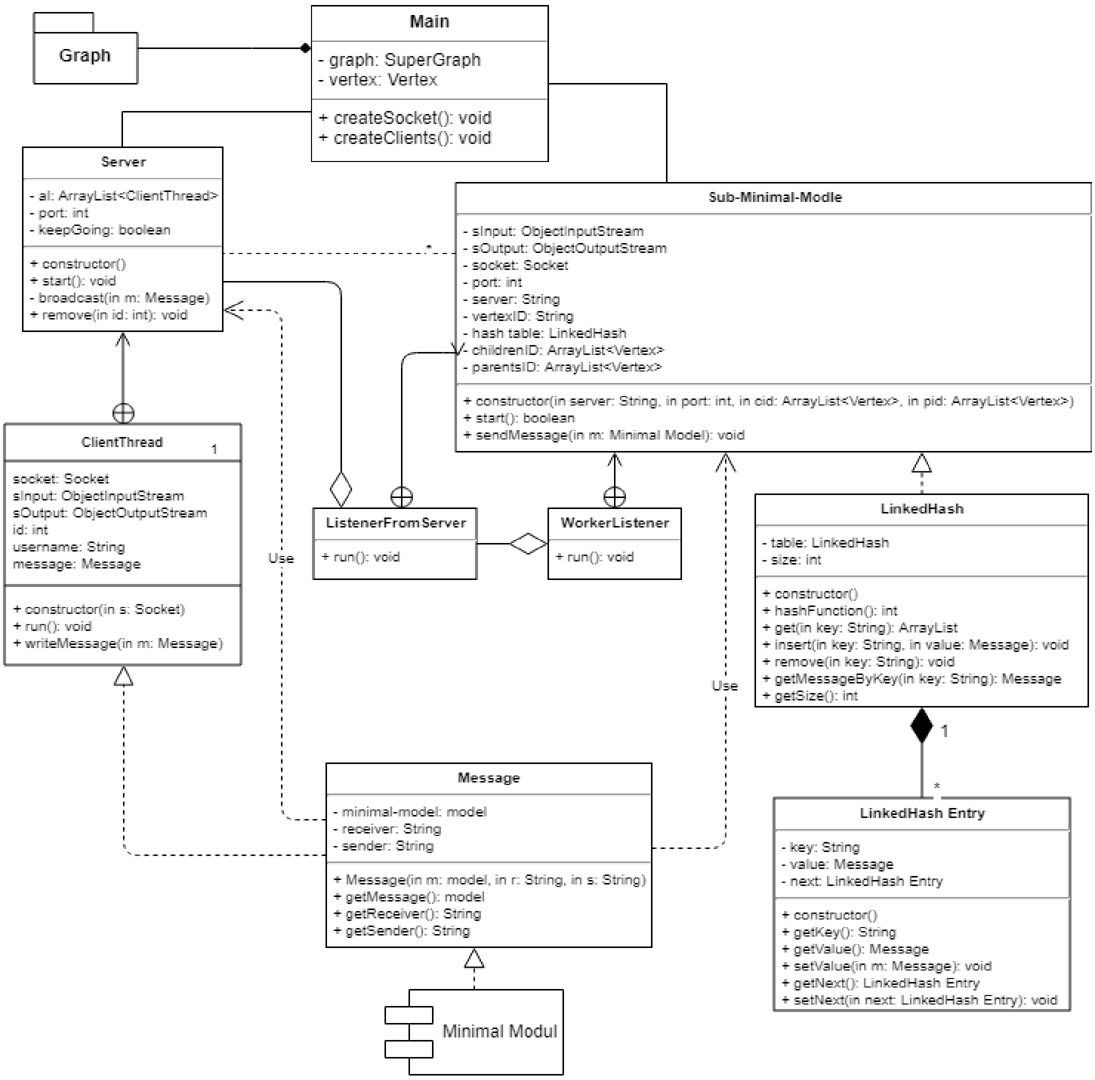
[1] R. Ben-Eliyahu-Zohary, F. Agiulli, F. Fassetti, L. Palopol, **Modular Construction of Minimal Models Artificial Intelligence**, 2014. http://dx.doi.org/10.1016/j.artint.2014.02.003
[2] R. Ben-Eliyahu-Zohary, R. Dechter, **On Computing Minimal Models**, 1994.
[3] R. Ben-Eliyahu-Zohary, C. Aviv, **An Upper Bound on Computing all X-minimal Models**
[4] R. Ben-Eliyahu-Zohary, **An Incremental Algorithm for Generation All Minimal Models**, 2005.
[5] Davis Putnam, https://www.jstor.org/stable/1970289?seq=1#page_scan_tab_contents
[6] M. Alviano, C. Dodaro, N. Leone, Francesco Ricca: **Advances in WASP**. Proceedings of LPNMR (2015).
[7] M. Alviano, C. Dodaro, W. Faber, N. Leone, Francesco Ricca: **WASP: A Native ASP Solver Based on Constraint Learning**. Proceedings of LPNMR (2013).
[8] M. Alviano, C. Dodaro, Francesco Ricca: **Comparing Alternative Solutions for Unfounded Set Propagation in ASP.** Proceedings of AI*IA (2013).
[9] Junichiro Fukuyama, **NP-completeness of the Planar Separator Problems**. Department of Mathematics and Computer Science Indiana State University, 2006.

7.2 תרשימים וטבלאות

באישור מנחה, מצורפים שני תרשימים המתארים את המערכת.

7.2.1 ארכיטקטורה

Class Diagram 7.2.2



7.3 תכנון הפרויקט – תאריכים משוערים

פגישת התנעה עם המנחה	22.10.18
סיום שלב התנעה	23.10.18
למידת נושא המחקר – מודל מינימלי ומודל מינימום	
פגישת הנחייה לקראת תחילת הפרויקט	27.11.18
הגשת טיוטת הצעה למנחה	15.12.18

16.12.18	מחקר וסקירת ספרות
19.12.18	פגישת הנחייה להצעת פרויקט והמשך המחקר
20.12.18	פגישת הנחייה להמשך המחקר
20.12.18	**סיום הגשת הצעת הפרויקט**
01.01.19	פגישת הנחיה לכתיבת האלגוריתם כפסאודו קוד
06.01.19	הגשת טיוטת פסאודו קוד והנחיה לקרת פיתוח ומימוש האלגוריתם
	למידת חישוב מבוזרת באמצעות תהליכונים
	פיתוח אלגוריתם מבוזר כבסיס למימוש האלגוריתם
18.02.19	הגשת טיוטת דוח אלפא למנחה

7.4 טבלת סיכונים

#	הסיכון	חומרה	מענה אפשרי
1	עמידה בזמנים – לא נספיק לפתח ולממש את האלגוריתם במועד.	בינוני	תכנון זמנים יעיל. קביעת ימי עבודה קבועים.
2	מכשולים תכנותיים – חוסר הצלחה בעת מימוש האלגוריתם.	בינוני	אעזר במנחה ובמידת הצורך בגורמים חיצוניים.
3	זמן ריצה – לא נשפר את זמן הריצה בהשוואה לאלגוריתמים קיימים.	גבוה	שימוש במבני נתונים שפעולות – הכנסה/מחיקה/חיפוש, הינן מהירות.
4	מכשולי תהליכונים, למשל deadlock, הרעבה וכו'..	נמוכה	בדיקות בהתאם לבעיה
5	עיכוב של תהליכי פיתוח בבניית מבני הנתונים.	גבוה	שימוש במבני נתונים קיימים.
6	עבור בדיקות עם מספר גדול של משתנים לא נמצא מודל מינימלי/מינימום	גבוה מאוד	לנסות לנתח את אופן החישוב ומימושו.
7	בחירת סביבת עבודה ושפת תכנות לא תואם הרצת סט גדול של חוקים.	בינוני	מחקר מקדים עבור הדרישות בפרויקט מול אפשרויות השפה.

7.5 רשימת\טבלת דרישות

7.5.1 דרישות מהאלגוריתם

מס'	סוג	תיאור דרישה
1	פלטפורמת מימוש	מימוש האלגוריתם בשפת Java, בסביבת eclipse.
2	אופן ביצוע	סנכרון בין תהליכים ועבודה מקבילית.
3	פונקציונאלי	שימוש במבנה נתונים דינאמי ויעיל.

4	אופן ביצוע	העברת הודעות בין תהליכונים.
5	אופן ביצוע	בניית גרף בהתאם לסט חוקים.
6	אפיון טכני	מבנה נתונים יעיל לתהליכון עבור שמירת הודעות שמתקבלות.
7	אפיון טכני	תקשורת בין תהליכונים במבנה שרת-לקוח.
8	אופן ביצוע	האלגוריתם יזהה את המודלים המינימליים הקיימים בתאוריה
9	אופן ביצוע	האלגוריתם יזהה מודלי מינימום הקיימים בתאוריה
11	פונקציונאלי	זמן ריצה יעיל ככל שניתן

7.5.2 דרישות מבחינת הנדסת תוכנה

מס'	תיאור דרישה
1	המערכת תבדוק אלגוריתמים למציאת מודל מינימום
2	תוצאת המערכת תכלול זמן ריצה וניצול זיכרון
3	המערכת תתמוך בממשק להזנת התאוריה
4	מימוש האלגוריתם באמצעות חישוב מבוזר

7.6 דוגמת הרצה – מציאת מודל מינימלי

T – סט של חוקים, תיאוריה.

M – המודל המינימלי.

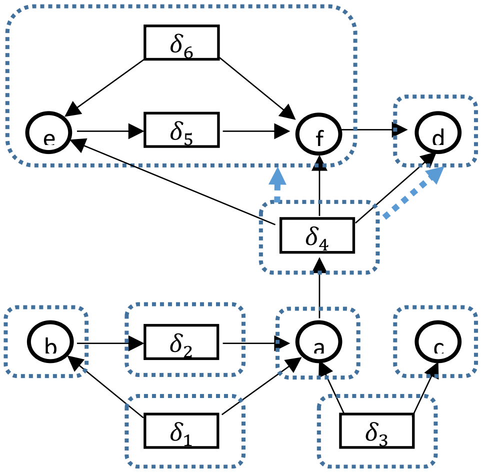
כל עוד T אינו קבוצה ריקה:

- מסט החוקים T, צור גרף של רכיבי קשירות.
- S – המקור של גרף רכיבי הקשירות.
- Ts – תת קבוצה של T המכילה את כל החוקים שיש בהם משתנים מS. ערכי המשתנים S יקבלו ערכי FALSE.
- X – מודל מינימלי עבור Ts.
- הוסף את המודל המינימלי שמצאת (X) ל-M.
- החסר מT את תת הקבוצה Ts ואת המודל המינימלי X.

```
Algorithm 1: Algorithm ModuMin
  Input: A positive theory T
  Output: A minimal model for T
M := ∅ ;
while T ≠ ∅ do
  if There is a clause δ in T violated by M such that |head(δ)| = 1 then
    let X := head(δ); M := M ∪ X ;
    T := Reduce(T, X, ∅) ;
  else
    let G be the super-dependency graph of T;
    Iteratively delete from G all the empty sources ;
    let S be the set of atoms in a source of G ;
    let T_S be the subset of T containing all the clauses from T having only
      atoms from S;
    let X be a minimal model of T_S;
    M := M ∪ X ;
    T := T − T_S; T := Reduce(T, X, S − X);
return M
```

T:

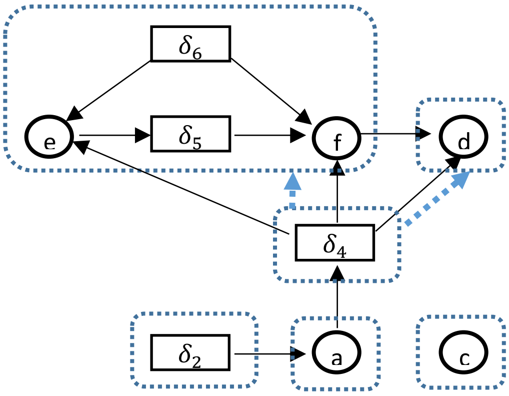
δ_1: $a \vee b$
δ_2: $b \rightarrow a$
δ_3: $a \vee c$
δ_4: $a \rightarrow d \vee e \vee f$
δ_5: $e \rightarrow f$
δ_6: $f \rightarrow e$



איור 1. [סופר] גרף תלויות של תיאוריה T

1.1. נבחר $source(b)$. $T(b) = \{\}$.
נעדכן את סט החוקים ואת הגרף כך ש$b = FALSE$.
$M = \{\}$.

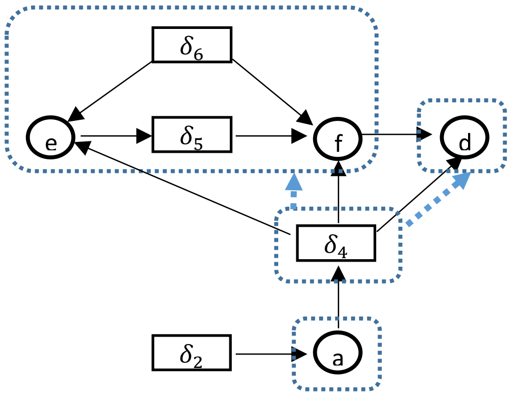
δ_1: $\cancel{a \vee b}$
δ_2: $\cancel{b} \rightarrow a$
δ_3: $a \vee c$
δ_4: $a \rightarrow d \vee e \vee f$
δ_5: $e \rightarrow f$
δ_6: $f \rightarrow e$



איור 1.1 [סופר] גרף תלויות של תיאוריה T

1.2. נבחר $source(c)$. $T(c) = \{\}$.
נעדכן את סט החוקים ואת הגרף כך ש- $c = FALSE$.
$M = \{\}$.

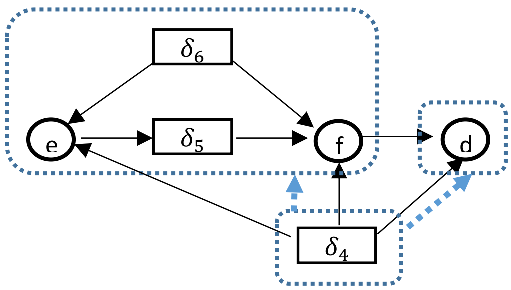
δ_1: $\cancel{a} \vee \cancel{b}$
δ_2: $\cancel{b} \rightarrow a$
δ_3: $\cancel{a} \vee \cancel{c}$
δ_4: $a \rightarrow d \vee e \vee f$
δ_5: $e \rightarrow f$
δ_6: $f \rightarrow e$



איור 1.2 [סופר] גרף תלויות של תיאוריה T

1.3. נבחר $source(a)$. $T(a) = \{a\}$.
נעדכן את סט החוקים ואת הגרף כך ש- $a = TRUE$.
נוסיף למודל המינימלי M את a. $M = \{a\}$.

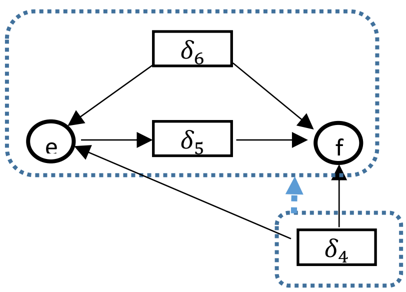
δ_1: $\cancel{a \vee b}$
δ_2: $\cancel{b \rightarrow a}$
δ_3: $\cancel{a \vee c}$
δ_4: $\cancel{a} \rightarrow d \vee e \vee f$
δ_5: $e \rightarrow f$
δ_6: $f \rightarrow e$



איור 1.3 [סופר] גרף תלויות של תיאוריה T

1.4. נבחר $source(d)$. $T(d) = \{\}$.
נעדכן את סט החוקים ואת הגרף כך ש- $d = FALSE$.
$M = \{a\}$.

δ_1: $\cancel{a \vee b}$
δ_2: $\cancel{b \rightarrow a}$
δ_3: $\cancel{a \vee c}$
δ_4: $\cancel{a} \rightarrow \cancel{d} \vee e \vee f$
δ_5: $e \rightarrow f$
δ_6: $f \rightarrow e$



איור 1.4 [סופר] גרף תלויות של תיאוריה T

Ramat Bet Hakerem, P.O.B 3566, Jerusalem 9103501 רמת בית הכרם, ת.ד. 3566, ירושלים מיקוד 9103501
Tel. 972-2-6588000 | Fax. +972-2-6588001 | Email: info@jce.ac.il | Web: www.jce.ac.il

1.5. נבחר $source(e,f)$. $T(e,f) = \{\{e \vee f\},\{e \to f\},\{f \to e\}\}$.
נעדכן את סט החוקים ואת הגרף כך ש$e = TRUE, f = TRUE$.
נוסיף למודל המינימלי M את e,f. $M = \{a,e,f\}$.

δ_1: $\cancel{a \vee b}$
δ_2: $\cancel{b \to a}$
δ_3: $\cancel{a \vee c}$
δ_4: $\cancel{a \to d \vee e \vee f}$
δ_5: $\cancel{e \to f}$
δ_6: $\cancel{f \to e}$

1.6. $T = \emptyset$. כלומר עברנו על כל סט החוקים.
המודל המינימלי שקיבלנו הוא: $M = \{a,e,f\}$.