

המחלקה להנדסת תוכנה

פרוייקט גמר – תשע"ח

חישוב מודלים מינימליים

Modular Construction Of Minimal Models

מאת: עמרי מזרחי

מנחה אקדמי: ד"ר יהודה חסין

אישור _____ תאריך _____

רכז הפרוייקטים: מר שפנייר אסף.

אישור _____ תאריך _____

עבודה משותפת עם הפרויקט של עדי טיירי

תוכן עניינים

2.....	מערכת ניהול הפרוייקט
3.....	תקציר
3.....	מילון מונחים, סימנים וקיצורים
4.....	מבוא
5.....	תיאור הבעיה
6-7.....	תיאור הפתרון
8-9.....	תרשימים וטבלאות
9.....	תכנון הפרוייקט
10.....	טבלת סיכונים
11.....	טבלת דרישות
12-14.....	סקירת ספרות/ביבליוגרפיה
12.....	עבודות דומות

מערכת ניהול הפרויקט

#	מערכת	מיקום
1	מאגר קוד	https://github.com/mazmaz2k/Modular-Construction-of-Minimal-Models.git
2	יומן	https://calendar.google.com/calendar/embed?src=k9htb14njhtete6mmgv i659at8%40group.calendar.google.com&ctz=Asia%2FJerusalem
3	ניהול פרוייקט	https://github.com/mazmaz2k/Modular-Construction-of-Minimal-Models/projects/1
4	הפצה	https://github.com/mazmaz2k/Modular-Construction-of-Minimal-Models/releases

תקציר

במסגרת פרויקט הגמר במחלקה להנדסת תוכנה בעזריאלי (JCE), הוטלה עלי המטלה לבצע פרויקט במסגרת של – 400 שעות

הפרויקט שאעשה הוא פרויקט מחקרי שנעשה בשיתוף פעולה עם הסטודנט עמרי מזרחי והמנחים שלנו פרופ' רחל בן אליהו זהרי וכן ד"ר יהודה חסין .

מטרת הפרויקט הוא לייעל אלגוריתם קיים למציאת מודלים מינימליים:

Modular Construction of Minimal Models by Rachel Ben-Eliyahu-Zohary, Fabrizio Angiulli, Fabio Fassett, and Luigi Palopoli. **לא נמצאו ערכי אינדקס.**

פרויקט זה הוא חלק אינטגרלי ונעשה בשיתוף פעולה מלא עם הפרויקט של עדי טיירי, כאשר בפרויקט של עדי נעסוק בחלק של מבנה הנתונים של הפסוקיות, בפרויקט זה נעסוק בחלק הגרפי של האלגוריתם, גרף זה נבנה ממבנה הנתונים של הפרויקט של עדי טיירי.

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

מילון מונחים , סימנים וקיצורים :

פורמט החוקים יהיה בצורה הבאה: $a_1 \wedge \dots \wedge a_n \rightarrow b_1 \vee \dots \vee b_m$

החלק שלפני החץ נקרא body והחלק שאחריו נקרא head .

מודל – השמה שמספקת את סט החוקים.

מבוא

בפרויקט זה נתייחס לבעיית ה-SAT או בעברית בעיית הספיקות שהיא בעיית הכרעה שהוכחה כ-NP שלמה (קוק-ליון) משמעות זו היא שלא קיימת לבעייה זו אלגוריתם שפותר אותה בזמן סביר.

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

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

מודל מינימלי

בהנחה ש m הוא מודל של תיאוריה מסוימת T , נגדיר $\text{positive}(m)$ להיות קבוצת המשתנים להם m מציבה True. נאמר ש M היא קבוצת כל המודלים אזי $m \in M$ הוא מודל מינימלי עבור T אם"ם לא קיים מודל $m' \in M$ כך ש $\text{positive}(m') \subset \text{positive}(m)$.

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

על מנת לבנות מודל מינימלי צריכים לבנות מסט החוקים הנ"ל גרף וקיים אלגוריתם שמוצא את המודל המינימלי בזמן ריצה אקספוננציאלי בגודל רכיב הקשירות הגדול ביותר של הגרף.

בפרויקט זה נמצא את רכיב הקשירות הגדול ביותר של הגרף וננסה לפרק אותו ע"י הצבת ערכים בוליאניים במשתנים ובכך להקטין את זמן הריצה של האלגוריתם הקיים.

תיאור הבעיה

קיים אלגוריתם שמוצא את המודל המינימלי עבור החוקים בזמן אקספוננציאלי בגודל רכיב הקשירות הגדול ביותר. יש מקרים עבור סט של חוקים בהם רכיב הקשירות בגרף הוא משמעותית גדול ולכן זמן הריצה יהיה לא קונבנציונלי.

חשוב מאוד שנשתמש באלגוריתמים יעילים כדי למצוא ולפרק את הרכיב הקשירות בכדי שנוכל להשתמש באלגוריתם הקיים באופן יותר יעיל מבחינת זמן הריצה שלו.

- פרויקט זה מסתעף לשני כיוונים בפתרון הבעיה הראשון הוא בניית החוקים במבנה נתונים דינמי בו ניתן לשנותו עבור השמה מסוימת של משתנה וחזרה אחורה במידה ויש צורך לשנות השמה והשני הוא בניית הגרף מציאת רכיב הקשירות הגדול וכן מציאת קדקוד חשוד לפירוק הרכיב.

בעיות בבניית מבני הנתונים לחוקים:

ישנה בעיה באופן בניית מבני הנתונים לחוקים כיצד לשמור את נתוני החוקים באופן יעיל כך שיהיה דינמי עבור שינוי השמה של משתנה איך לשלוח נתונים מהחוקים אל הגרף ובחזרה.

כאשר נרצה לבצע השמה בוליאנית של משתנה מבנה הנתונים יצטרך לתמוך במחיקה של משתנה או של פסוקית שלמה בהתאם לחוקי הלוגיקה של סט החוקים.

מבנה הנתונים של החוקים אף יצטרך לתמוך ב backtracking על מנת לשחזר סט קיים שהצבנו בו כבר ערכים

וכן עבור מספר משתנים שבחרנו לבצע עליהם השמה יש לבדוק האם עבור השמה שאנו רוצים להשתמש בה בכלל נקבל מודל או שנצטרך לבדוק עבור אותם המשתנים ערכים שונים ז"א אם יש לנו n משתנים אנו נצטרך לבדוק 2^n אפשרויות בהם יש 2 בדיקות לכל השמה, הראשונה היא האם קיבלנו פירוק יעיל לרכיב הקשירות הגדול ביותר והשנייה האם עבור השמה זו יהיה לנו מודל.

בעיות בבניית מבני הנתונים לגרף:

בניית הגרף צריכה להתבצע בהתאם למבנה הנתונים של החוקים ובהתאם ללוגיקה של סט החוקים. יש למצוא את רכיב הקשירות הגדול ביותר בגרף ולבצע עליו מניפולציות כך שיהיה ניתן לדעת איזה קודקוד(משתנה) הוא חשוד לפירוק רכיב הקשירות.

המחלקה של מבנה הנתונים של הגרף צריכה להיות בקשר באמצעות ממשק עם המחלקה של החוקים כך שיהיה ניתן לשלוח קודקוד עליו ניתן לבצע השמה ולקבל בחזרה מהחוקים אילו קודקודים/פסוקיות אותם צריך למחוק מהגרף.

בעיה בבדיקות

האם לאחר פירוק רכיב הקשירות ושליחת סט החוקים החדש לאלגוריתם הקיים זמן הריצה אכן קטן מזמן הריצה של סט החוקים הישן.

תיאור הפתרון

כאמור בפרויקט זה נתרכז בפתרון בעיית הגרף אך מכיוון שפרויקטים אלה הולכים יד ביד נתאר גם את חלק מפתרון בעיית מבנה הנתונים של החוקים שניתן רקע לפתרון בעיית הגרף.

בקבלת קובץ עם סט חוקים נחלץ את כל נתוני הקובץ למחלקת החוקים ונשמור אותם במבנה נתונים דינמי. מבנה נתונים זה יהיה מבנה הנתונים של החוקים.

הגרף יכול לתת עד שלושה משתנים שכאשר נעשה להם הצבה של ערכים True/False נקבל רשימת חוקים קטנה שעליה יהיה ניתן להריץ את האלגוריתם למציאת מודל מינימאלי.

פתרונות במחלקות הגרף:

מחלקת החוקים תשלח את מבנה הנתונים באופן ישיר למחלקת הגרף במטרה לבנות את הגרף.

כל משתנה במבנה הנתונים של החוקים יהיה קודקוד וכל חוק יהיה מספק קודקודים שמחוברים יחדיו בקשתות.

נבנה את הגרף בכך שנחזיק מערך שבו יש את כל הקודקודים ומספר החוקים שבהם הוא מופיע, זה בשביל שתהיה לנו שפה משותפת עם מבנה הנתונים של החוקים כי כך נוכל למחוק קודקוד ולמחוק חוק.

נבנה מערך שבו נחזיק את רשימת הקשתות, כל קשת תחזיק קודקוד מוצא וקודקוד יעד ואיזה חוק קשת זאת שייכת.

לאחר בניית הגרף לפי סט החוקים הנתון לנו יש למצוא את רכיב הקשירות הגדול ביותר לצורך זה נשתמש באלגוריתם DFS הפותר זאת בזמן לינארי בגודל הגרף $O(V+E)$ נמצא את רכיבי הקשירות ונחפש את רכיב הקשירות עם הכי הרבה קודקודים לפי גרף ה SCCs (גרף רכיבי הקשירות)

עכשיו כשמצאנו את הרכיב הקשירות הגדול ביותר בגרף נשתמש באלגוריתם למציאת קודקוד שיפרק את הרכיב ל-2 חלקים באמצעות Strong articulation point algorithm. אלגוריתם זה נותן לנו קודקודים חשודים ברכיב הקשירות באמצעותם יהיה ניתן לפרק את הגרף.

ברגע שמצאנו את הקודקודים שיכולים לפרק את רכיב הקשירות נשלח אותם למחלקת החוקים בכדי להציב אותם ולשנות את מבנה הנתונים .

- האלגוריתם הקיים מוצא מודל מינימלי לסט החוקים בזמן ריצה אקספוננציאלי בגודל רכיב הקשירות הגדול ביותר, זמן הריצה לאחר השיפור שלנו נשאר אקספוננציאלי בגודל רכיב הקשירות הגדול ביותר אבל רכיב הקשירות קטן משמעותית בעקבות פירוקו ולכן זמן הריצה יקטן בהתאם לטיב הפירוק.

תיאור הכלים המשמשים לפתרון

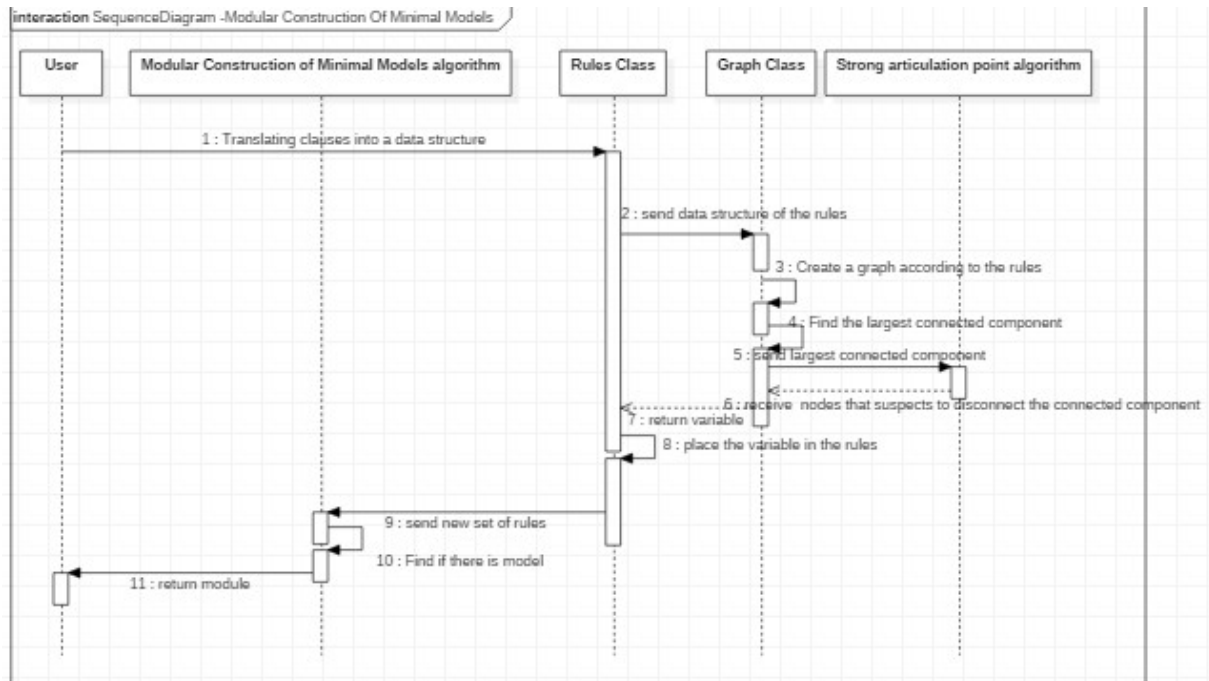
בפרויקט נשתמש במסגרת הפיתוח של Eclipse כדי לפתח את התוכנית שלנו.
את סביבת הפיתוח אנו נריץ במערכות הפעלה Windows ושפת התכנות איתה נעבוד תהיה Java .

הפרויקט ינוהל לחלוטין ע"י מערכת GitHub ננהל שם את:

1. מאגר הקוד
2. יומן אירועים
3. לוח מטלות
4. מצב הפרויקט ולוח ניהול לפרויקט

נספחים:

א. תרשימים וטבלאות:



Sequence diagram:

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.

מחלקת מבנה הנתונים של החוקים ישלח את מבנה הנתונים של החוקים למחלקת מבנה הנתונים של הגרף.

ניצור גרף שייצג את כל החוקים. כאשר נסיים את בניית הגרף נמצא את רכיב הקשירות הגדול ביותר.

את רכיב הקשירות הגדול ביותר נשלח למחלקת Strong Bridges and Strong Articulation Points of Directed Graph.

האלגוריתם של המחלקה הנ"ל ימצא לנו קודקודים החשודים לפירוק רכיב הקשירות.
נחזיר את הקודקודים למחלקת הגרף ומחלקת הגרף תחזיר את משתנים שמתאימים לקודקודים למחלקת מבנה הנתונים של החוקים.
מחלקת החוקים תציב True/False בקודקודים שבתוך סט החוקים, וכך נקבל מודל מצומצם יותר.
לאחר מכן נריץ את האלגוריתם למציאת מודל מינימאלי על סט החוקים לאחר ההצבה.

ב. 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
21.1	שלב האב טיפוס - מימוש, מחקר, סקירת ספרות, ארכיטקטורה ובדיקות
20.4	שלב הבנייה - הגשת מסמך תיכון ומימוש
[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

#	הסיכון	חומרה	מענה אפשרי
1	עיכוב של תהליכי פיתוח בבניית מבני הנתונים.	4	שימוש במבני נתונים קיימים אשר נמצאים באינטרנט.
2	האלגוריתם לפירוק רכיב הקשירות הגדול ביותר לא יפרק את רכיב הקשירות בצורה שווה	2	ננסה לפרק באמצעות כמה קודקודים . עדיין נשפר את זמן הריצה לאחר הפירוק.
3	לא נמצא סט גדול של חוקים שנדע את זמן הריצה הלא יעיל לפתרון שלו	6	נפתור עם סט חוקים קיימים שיש לנו כבר.
4	הסנכרון של מחלקת הגרף ומחלקת מבנה הנתונים של החוקים לא יעבוד	9	העבודה על בניית המחלקות תעשה בתיאום מוחלט.
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 יעילות
6	ניסיון השחזור לפי קובץ ה-log יתברר כלא יעיל	5	נמצא פתרון יותר אלגנטי ברמת הקוד לזיכרון של קודקודים וחוקים שנמחקו
7	אם לא קיים מודל מינימלי לאחר פירוק רכיב הקשירות	4	אם לא יהיה מודל מינימלי נשתמש במודל הרגיל.

ד. רשימת/טבלת דרישות

טבלת דרישות (User Requirement Document)

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 head → body
7	אופן ביצוע	המערכת תתחזק מבנה נתונים השומר עבור כל משתנה באיזה חוק הוא נמצא
8	פונקציונאלי	המערכת תריץ את האלגוריתם Strong Bridges and Strong Articulation Points of directed Graph.
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.

סקירת ספרות/ביבליוגרפיה ועבודות דומות

עבודות דומות:

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)

Davis Putnam:

דוגמה לעבודה דומה למה שאנו עושים בפרויקט זה הוא האלגוריתם של Davis Putnam שזה אלגוריתם שמוצא מודל מינימלי בכך שהוא מתעדף בהצבה של במשתנים 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. ראה ביבליוגרפיה סעיף 34.

סקירת ספרות/ביבליוגרפיה :

1. F. Angiulli, R. Ben-Eliyahu-Zohary, F. Fasseti, and L. Palopoli. On the tractability of minimal model computation for some cnf theories. *Artificial 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 Mathematics and Artificial Intelligence*, 18:3-27, 1996.
4. R. Ben-Eliyahu-Zohary. An incremental algorithm for generating all minimal models. *Artificial Intelligence*, 169(1):1-22, 2005.
5. R. Ben-Eliyahu-Zohary and L. Palopoli. Reasoning with minimal models: Efficient algorithms and applications. *Artificial Intelligence*, 96(2):421-449, 1997.
6. N. Bidoit and C. Froidevaux. Minimalism subsumes default logic and circumscription in stratified 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 finding for nonmonotonic propositional logics. In *Proceedings of the 4th Italian conference on theoretical computer science*, pages 125-139. World Scientific 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 Ben-Eliyahu-Zohary et al.
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 Proceedings 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. *Artificial Intelligence*, 151(1):177-212, 2003.
24. P. G. Kolaitis and C. H. Papadimitriou. Some computational aspects of circumscription. *J. ACM*, 37(1):14, 1990.
25. N. Leone, P. Rullo, and F. Scarcello. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. *Inf. Comput.*, 135(2):69-112, 1997.
26. V. Lifschitz. Computing circumscription. In *IJCAI-85: Proceedings of the international 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. *Artificial Intelligence*, 13:27-39, 1980.
29. J. McCarthy. Application of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28:89-116, 1986.
30. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81-132, 1980.
31. P. Simons, I. Niemela, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138(1):181-234, 2002.
32. R. T. Stern, M. Kalech, A. Feldman, and G. M. Provan. Exploring the duality in conflict-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