**הסקה אוטומטית – תרגיל 2**

**נתונים טכניים**

* תאריך ההגשה לתרגיל 25.12.2024
* ההגשה בזוגות או לבד.
* ההגשה תתבצע במערכת הלמדה על ידי אחד או אחת מהזוג.
* יש להגיש קובץ zip הכולל: את כל קבצי הקוד, קובץ דוח pdf עם שמות ות.ז, כל התשובות לשאלות שבתרגיל ותיקיית benchmark של חלק א.
* בחלק התכנותי ניתן ואף מומלץ להריץ ולבדוק יותר מקרים ממה שחייבים ולהוסיף לדוח את המסקנות.

**חלק א - אסטרטגיות ויוריסטיקות בחירה**

בכיתה למדנו את תחשיב CDCL. בתרגיל זה נרצה לממש ולבחון את הייעילות של אסטרטגיות שונות והוספת יוריסטיקות בחירה לתחשיב CDCL. בתרגיל תצטרכו לכתוב דוח מפורט על הפותרנים השונים עם הסבר על המימוש ותוצאות אמפיריות המשוות בין הפותרנים השונים.

בתרגיל זה יש לממש 4 פותרנים שונים המבוססים על תחשיב CDCL. יש לבחור 2 אסטרטגיות שונות ו-2 יוריסטיקות בחירה וליצור מהם 4 פותרנים:

* פותרן עם אסטרטגיה א ויוריסטיקה א
* פותרן עם אסטרטגיה א ויוריסטיקה ב
* פותרן עם אסטרטגיה ב ויוריסטיקה א
* פותרן עם אסטרטגיה ב ויוריסטיקה ב

בחרו 2 אסטרטגיות לביצוע תחשיב CDCL (שונות ממה שהיה בתרגול). יש להסביר עבור כל אסטרטגיה מדוע אסטרגטגיה זו תועיל.

בחרו 2 יוריסטיקות לבחירת משתנה בכלל decide על פי הנלמד בתרגול, אחת מהיוריסטיקות חייבת להיות VSID (ואי אפשר לבחור את DLIS).

יש ליצור 100 קבצי cnf על מנת להשוות בין הפותרנים השונים (שמות הקבצים יהיו formula\_1.cnf, … , formula\_100.cnf). כל קובץ יכיל 50 משתנים ובכל פסוקית יהיו 3 ליטרלים בלבד. יצירת קבצים אלו יהיו לפי עקרונות phase transition שראינו בתרגול על מנת שיהוו סט של בעיות מעניינות (שימו לב שמשתנה לא מופיע פעמיים באותו פסוקיצ וכן אין פסוקיות חוזרות). כל הקבצים צריכים להיות שמורים בתיקייה benchmark.

בדו״ח יש להשוות בין ארבעת הפותרנים ביחד עם minisat והפותרן שקיבלתם בתרגיל וולהראות כמה בעיות נפתרו מתוך סט הבעיות שייצרתם עבור כל פותרן. יש להסביר את התוצאות ומסקנות העולות מתוך השוואות אלו.

כל פותרן צריך לרוץ עם timeot של 5 שניות לכל נוסחה כאשר אם הפותרן לא הצליח לפתור את הנוסחה בזמן הנתון הפותרן יחזיר unknown. לנוחיותכם מצורף סקריפט run\_solver.sh אשר מריץ פותרן על סט של נוסחאות עם timeout לבחירתכם וכותב את התוצאות לקובץ בשם res (בכל שורה יש תשובה של הפותרן לנוסחה אחת כך שבשורה ה-i יש את תשובת הפותרן לנוסחה furmula\_i.cnf). שימוש בסקריפט יתבצע בצורה הבאה:

run\_solver.sh <solver> <directory> <timeout>

כאשר הארגומנטים לפי הסדר הם: נתיב לפותרן, תיקייה של קבצי cnf, מספר שניות ל-timeout.

הריצו את minisat ואת הפותרן המצורף בתיקיית התרגיל בעזרת הסקריפט (הנתיב ל-minisat הוא פשוט minisat) והשוו את התשובות של הפותרנים שלכם לתשובות של minisat והפותרן המצורף. לאחר הרצת מספר פותרנים ניתן להשוות בין שני פותרנים בעזרת הפקודה:

diff -y --suppress-common-lines <(nl -ba <file1>) <(nl -ba <file2>)

כאשר file1, file2 אלו הקבצים של התשובות שביניהם אתם רוצים להשוות, שימו לב שהפותרנים מסכימים ביניהם ולא מחזירים תשובות הפוכות. ניתן גם למצוא את כמות השורות שבהם יש הבדל בין שני הקבצים בעזרת הפקודה:

diff -y --suppress-common-lines <(nl -ba <file1>) <(nl -ba <file2>) | wc -l

**חלק ב – שאלות תאורטיות**

1. שאלות בלוגיקה פסוקית

א. לכל נוסחה פסוקית X נגדיר ב-(var(X את קבוצת המשתנים הפסוקיים המופיעים ב-X. לדוגמה:

var(p ) = {p,q}

תהיינה A ו-B נוסחאות בלוגיקה פסוקית כך שמתקיים:

var(A) var(B) = {}

נניח כי הנוסחה תקפה. הוכיחו כי תקפה או תקפה.

ב. שלישייה (a,b,c) של מספרים שלמים וחיוביים נקראת שלשה פיתגורית אם . צביעה של המספרים {1n, …, n } היא פונקציה מ- {1n, …, n } ל-{blue,red}. שלישייה של מספרים (a,b,c) נקראת מונוכרומטית ביחס ל-f אם

f(a) = f(b) = f(c)

לדוגמה: אם n =10 ו-f היא הפונקציה שמחזירה תמיד res, אז השלישייה 3,4,5 היא שלישייה פיתגורית מונוכרומטית. כתבו נוסחה בלוגיקה פסוקית מעל המשתנים

{} שספיקה אם ורק אם קיימת צביעה f של {40,...., 1} ללא שלישייה פיתגורית מונוכרומטית ביחס ל-f. הציגו השמה שמספקת אותה, וכן צביעה ללא שלישייה פיתגורית מונוכרומטית שמתאימה להשמה.[[1]](#footnote-0)

ג. הוכיחו כי בעיית הספיקות עבור נוסחאות dual-Horn היא ב-P. תוכלו להסתמך על המשפט שהוכחנו לגבי נוסחאות Horn, אך אין חובה לעשות זאת.

1. הוכיחו או הפריכו לגבי תחשיב DPLL:

א. לכל קונפיגורציה (M, F, D), אין משתנה שמופיע פעמיים ב-M.

ב. לכל קונפיגורציה (M, F, D), אם (M, F, D) גזירה מ-( ,F ,[]) בתחשיב DPLL אז אין משתנה שמופיע פעמיים ב-M.

ג. לכל קונפיגורציה (M, F, D), אין משתנה שמופיע ב-M אך לא ב-F.

ד. לכל קונפיגורציה (M, F, D), אם (M, F, D) גזירה מ-( ,F ,[]) בתחשיב DPLL אז אין משתנה שמופיע ב-M אך לא ב-F.

ה. לכל קונפיגורציה (M, F, D), אין משתנה שמופיע ב-F אך לא ב-M.

ו. לכל קונפיגורציה (M, F, D), אם (M, F, D) גזירה מ-( ,F ,[]) בתחשיב DPLL אז אין משתנה שצופיע ב-F אך לא ב-M.

1. לקריאה נוספת: <https://www.cs.utexas.edu/~marijn/ptn/> [↑](#footnote-ref-0)