# מבוא לאימות תוכנה – תרגיל בית 5

**חורף תשפ"ג**

מתרגל אחראי על התרגיל: רועי דויטש

שימו לב: תשובה לא מנומקת לא תזכה בנקודות.

עבור שאלות 1 ו-2, הניחו שקיימת פונקציה $SatSolve(\alpha)$ המקבלת נוסחה בוליאנית $\alpha$ (לא בהכרח ב-CNF) ומחזירה SAT אם $\alpha$ ספיקה ו-UNSAT אם $\alpha$ אינה ספיקה.
האלגוריתמים צריכים להיות יעילים ככל האפשר,
שימו לב, יעיל משמעותו מספר איטרציות מינימלי.

**שאלה 1**

נתון מבנה קריפקה $M = (S, I, R, L)$ מעל נוסחאות אטומיות $AP$ וקבוצת ***קשתות חולות*** $R_{ill} \subseteq R$.
נאמר כי מסלול $\pi = s_0, s_1, \ldots$ במבנה ***מכבד*** את $R_{ill}$ אם לכל קשת חולה, עוברים בה לאורך המסלול לכל היותר פעם אחת, כלומר, לכל מעבר $(s,t) \in R_{ill}$ מתקיים כי לא קיימים $i \neq j$ כך ש: $s_i = s, s_{i+1} = t$ וגם $s_j = s, s_{j+1} = t$.

נתונות הנוסחאות הפסוקיות הבאות, במבנה CNF, מעל וקטור משני המצב $\bar{v} = (v_1, \ldots, v_n)$:

- $S(\bar{v})$: מייצגת את קבוצת המצבים במבנה $M$.
- $I(\bar{v})$: מייצגת את קבוצת המצבים ההתחלתיים במבנה $M$.
- $R(\bar{v}, \bar{u})$: מייצגת את רלציית המעברים במבנה $M$.
- $R_{ill}(\bar{v}, \bar{u})$: מייצגת את רלציית המעברים החולים במבנה $M$.
  - נתון כי: $R_{ill}(\bar{v}, \bar{u}) \rightarrow R(\bar{v}, \bar{u})$
- לכל $p \in AP$, $p(\bar{v})$ מייצגת את קבוצת המצבים במבנה $M$ שמספקים את $p$.

א. נאמר כי מעבר $(s,t) \in R$ הוא בריא אם הוא אינו חולה.

כתבו נוסחה פסוקית $R_{healthy}(\bar{v}, \bar{u})$ שמייצגת את קבוצת המעברים הבריאים.
נמקו את נכונות הנוסחה.

ב. בסעיף זה הניחו כי הנוסחה שנתתם בסעיף הקודם נכונה וכי היא נתונה לכם.

תהא $p \in AP$. כתבו אלגוריתם איטרטיבי בסגנון BMC, יעיל ככל האפשר, המחזיר $True$ אם קיים במבנה מסלול $\pi$ שמכבד את $R_{ill}$ ומקיים $M, \pi \models Gp$ ו-$False$ אחרת.

הניחו כי נתונה פונקציה $SolveSat(\varphi)$ אשר בהינתן נוסחה פסוקית מחזירה *SAT* אם הנוסחה ספיקה, ו-*UNSAT* אחרת.

**שאלה 2**

נתון מבנה קריפקה המיוצג ע"י נוסחאות CNF מעל וקטור המשתנים $\bar{v} = (v_1,\dots,v_n)$:

- $S(\bar{v})$ - נוסחת CNF המייצגת את קבוצת המצבים של M.
- $I(\bar{v})$ - נוסחת CNF המייצגת את קבוצת המצבים ההתחלתיים של M.
- $R(\bar{v},\bar{v}')$ - נוסחת CNF המייצגת את רלציית המעברים של M.
- $p(\bar{v})$ - נוסחת CNF המייצגת את קבוצת המצבים המספקים את הנוסחה האטומית $p$.
- $q(\bar{v})$ - נוסחת CNF המייצגת את קבוצת המצבים המספקים את הנוסחה האטומית $q$.

הציעו אלגוריתם איטרטיבי בסגנון BMC יעיל ככל האפשר, אשר מחזיר $true$ אם קיים מצב התחלתי $s_0 \in I$ כך ש-$M, s_0 \models EG(EXp \wedge EXq)$, ו-$false$ אחרת.

# חלק לח (רטוב, אבל רק קצת)

כפי ש(בוודאי) ידוע לכם, לאחרונה הוקם "פורום העתודאים הארצי", ששם לעצמו מטרה לסייע לכלל העתודאים בישראל (לפרטים נוספים: https://forumatuda.wordpress.com). מחלקת מבצעים מיוחדים של הפורום קיבלה על עצמה את אחת המשימות המשמעותיות ביותר של הארגון - הפחתת מספר שנות שירות הקבע של העתודאים. הפרויקט משך עניין רב בפורום, ובמאמץ רב הושג קוד המקור של מערכת בקרת הכניסה של מדור עתודה (hack_me.c) המצורף לתרגיל זה. בתרגיל, ננסה לפרוץ את מערכת בקרת הכניסה הנתונה, על-מנת שדרכה נוכל לשנות את הנתונים הנדרשים במסד הנתונים של מדור עתודה.

למחקר המערכת גויסו סטודנטים מהפקולטה למדעי המחשב בטכניון ממגוון תחומים. בעוד שיוצאי הקורס "הגנה ברשתות מתוכנתות" מתכננים תקיפות על הקוד, בוגרי הקורס ב-RE מנסים לעשות דה-קומפילציה לקוד מקור ויוצאי הקורס "תכנות מקבילי ומבוזר" מדבגים את הניסיונות שלהם לפרוץ למערכת באמצעות מספר רב של שרתים, נתבקשתם אתם - סטודנטים בקורס במבוא לאימות תוכנה - להשתמש בכלי ה-CBMC לפיצוח המערכת.

1. בשלב זה נרצה להכיר את המערכת. ראשית, התבוננו בקוד הנתון. כפי שאתם רואים, בוגרי קורס מת"מ (שעיקר הלקחים ממנו הוא ידע מוגזם ב-define-ים) שינוי אותו והפכו אותו לבלתי-נסבל. בתרגיל אין צורך לפענח או להבין מה מבצעת הפונקציה verify. יש להתייחס אליה כאל קופסא שחורה המחשב פונקציה כלשהי. מעבר לכך, **פתרונות של התרגיל שיסתמכו על הבנה של האלגוריתם שמבצעת הפונקציה verify, מעבר למה שתואר בפסקה זו, לא יתקבלו.**

   התבוננו במבנה הכללי של הקוד, ונסו לקמפל ולהריץ אותו. שימו לב, בסעיף זה אין לשנות את הקוד כלל. הבינו וכתבו בקצרה כיצד יש להשתמש במערכת, כיצד מזינים לה קלט, ומהו המבנה של קלט תקין.

2. ידוע כי בהרשמה למערכת, המשתמש החדש בוחר לו שם משתמש, ובתגובה ניתנת לו סיסמת התחברות. בסעיף זה, נרצה למצוא סיסמת התחברות עבור שמות משתמש שונים. ספציפית, עבור כל הגשה של תרגיל בית זה, נתייחס לתעודת זהות של כל אחד מהמגישים כ-username אשר נרצה למצוא את הסיסמא עבורו (למשל, אם מגישים את התרגיל דניאל קימל, ת"ז 123456789, וליעם זולוטניצקי, ת"ז 987654321, נרצה למצוא את הסיסמאות המתאימות ל-123456789 ול-987654321). עליכם להשתמש ב-CBMC (הנחיות לשימוש - בהמשך מסמך זה) על מנת למצוא סיסמאות מתאימות.

   עבור פתרון סעיף זה, עליכם לשנות את הקובץ hack_me.c, כאשר מותר לשנות אך ורק קוד שרשום בפונקציית ה-main. בפרט, אין לשנות את הפונקציה verify. נדגיש שוב כי עליכם להשתמש ב-CBMC בלבד בתרגיל בית זה, ובפרט ביכולות ה-SAT SOLVER שבו, וכל השיטות שכרוכות בפענוח הקוד אסורות לשימוש.
   עליכם להציג את השינויים שביצעתם בקוד (כלומר – להציג בהגשה את חלקי הקוד ששיניתם. אין צורך להציג חלקי קוד שהופיעו בתרגיל המקורי). עליכם להסביר מהם השינויים שביצעתם, וכיצד הם סייעו לכם לפתור את הסעיף.

3. לצוות הפרויקט נודע כעת כי מעבר לסיסמאות האישיות לכל משתמש, קיימת "סיסמת מאסטר" - סיסמא אשר מתאימה לכל המשתמשים. עליכם למצוא אותה תוך שימוש ב-CBMC. ההגבלות של סעיף ב' חלות גם כאן.

   שימו לב: לסעיף זה שני חלקים. עליכם בתחילה **למצוא** סיסמא נוספת, שונה מזו שמצאתם בסעיף ב', שמתאימה לכלל המשתמשים. לאחר מכן, עליכם **להוכיח** באמצעות שימוש ב-CBMC שהיא אכן סיסמת מאסטר, כלומר – כזו שמתאימה לכל משתמש.
   עליכם להציג את השינויים שביצעתם בקוד (כלומר – להציג בהגשה את חלקי הקוד ששיניתם. אין צורך להציג חלקי קוד שהופיעו בתרגיל המקורי). עליכם להסביר מהם השינויים שביצעתם, וכיצד הם סייעו לכם לפתור את הסעיף.

4. את הסיסמאות שהשגתם בסעיף הקודם יש להגיש בפורמט csv בקובץ בשם passwords.csv בפורמט הבא:

```
Id1, password1
Id2, password2
master, master_password
```

5. במדור עתודה הבינו שנפרצה המערכת (חלה עלייה ניכרת בשיעור העתודאים המחייכים), גויסתם לעזרת המדור לצורך שדרוג המערכת. באחד המודולים של המערכת, מבצע המדור חיפוש בינארי במערך, ע"י הפונקציה הנתונה בקובץ binsearch.c. אתם נדרשים להשתמש בפונקציונליות שמציע CBMC על-מנת לבדוק האם הגישות למערך בפונקציה הנוכחית חוקיות.
   a. הדגל המתאים לבדיקה הנ"ל הוא bounds-check--. נסו להריץ את CBMC עם דגל זה בלבד על הפונקציה שבתכנית (היעזרו בדגל function--). מה הייתה התוצאה שהתקבלה? השוו הרצה זו להרצה של הקובץ predictable.c עם אותם דגלים.
   בונוס: הציעו הסבר אפשרי להבדל.
   b. נסו להריץ את הבדיקה עם unwinding של 1 (unwind 1--). מהי התוצאה? היעזרו בדגל no-unwinding-assertions-- על-מנת לבדוק האם האימות עם החסם הנוכחי על הפרישה הצליח או לא. מהי משמעות הדגל? מה ניתן להסיק מהרצה זו לגבי נכונות התכנית?
   c. נסו כעת להגדיל את החסם על ה-unwinding. מהי התוצאה? מה ניתן להסיק מההרצה החדשה לגבי נכונות התכנית?

## CBMC – הנחיות שימוש

כלי ה-CBMC מותקן בשרת ה-csl3, אליו יש לכם גישה באמצעות פרטי ההתחברות הטכניונית. הכלי נמצא ב:

usr/local/cbmc/cbmc/

מומלץ לבצע alias על-מנת להקל על השימוש:

alias cbmc=/usr/local/cbmc/cbmc

על-מנת להריץ את הכלי על קובץ c בשם example.c, יש להזין:

cbmc example.c

להנחיות נוספות, ניתן להסתייע ב:

- הזינו cbmc --help.
- שקפי התרגול.
- גוגל.

שימו לב כי הכלי ניתן לשימוש גם בפלטפורמות נוספות, כפי שמצוין בתרגול.

**בהצלחה!**