

שיטות וכלים לשיפור יעילות, סמס' א' תשע"ז, בחינה לדוגמא

פרופ' מולי שגיב, כלב אלפרנס

הנחיות כלליות:

- משך הבחינה 3 שעות.
- מותר להשתמש בכל חומר כתוב (אין להשתמש במחשבון, מחשב, או טלפון סלולרי).
- במבחן 6 שאלות. יש לענות על 5 שאלות בלבד.
- את כל התשובות יש לכתוב במחברת הבחינה בלבד, תשובות על טופס הבחינה לא ייבדקו.
- חלוקת הניקוד: 20 נקודות לכל שאלה. סה"כ 100 נקודות.

SAT .1

סעיפי השאלה מתייחסים לנוסחה הבאה:

$$(x1 \vee \neg x2 \vee \neg x3) \wedge (\neg x1 \vee \neg x2 \vee \neg x3) \wedge (x2 \vee x3) \wedge (x3 \vee x4) \wedge (x3 \vee \neg x4)$$

- הראי הפעלה של תהליך הרזולוציה (resolution) על הנוסחה.
- האם הנוסחה ספיקה? אם כן, הראי השמה מספקת לנוסחה.
- הראי הפעלה של תהליך DPLL על הנוסחה.

Bounded Model Checking .2

תני דוגמא לתוכנית עם שגיאה של חלוקה באפס שאינה מתגלה על ידי כלי BMC בגלל חסם נמוך מידי.

Concolic Execution .3

הסבירי אילו קריאות ל SMT Solver יתבצעו בעת הרצת כלי Concolic Execution על הפונקציה test בתוכנית הבאה:

```
int half(int v) { return v/2; }

void test(int x, int y)
{
  z = half(y);
  if (x == z)
  {
    if (x > y + 15)
      ERROR;
  }
}
```

Alloy .4

מודל Alloy שלפנייך מתאר עץ משפחה.

- א. האם במודל זה ייתכן שאישה א' נשואה לאישה ב' אך אישה ב' לא נשואה לאישה א'?
- ב. כתבי פרדיקט (predicate) שבודק האם תיתכן התכונה מסעיף א'.
- ג. כתבי עובדה (fact) שתתקן את המודל, כך שלא ייתכן אדם א' (גבר או אשה) שנשוי לאדם ב', בעוד אדם ב' אינו נשוי לאדם א'.

```
abstract sig Person {
  father: lone Man,
  mother: lone Woman,
  spouse: lone Person
}

sig Man extends Person {}

sig Woman extends Person {}

fact Biology { no p: Person | p in p.^(mother+father) }
```

Deductive Verification .5

הוכיחי את נכונות התוכנית הבאה באמצעות Hoare Style Verification. שימי לב שעלייך להגדיר תנאי קדם (pre) ובתר (post) נכונים לתוכנית, וכן אינוואריאנטה אינדוקטיבית, ולהוכיח שהתוכנית מקיימת אותם.

```
def sum(a: int[], k: int) returns (s: int):
  s := 0
  i := 0
  while (i < k) do:
    s := s + a[i]
    i := i + 1
  return s
```

6. Abstract Interpretation

הפעילי את אלגוריתם Chaotic Iteration על התוכנית שלפניך, בדומיין האבסטרקטי Available Expressions, ששומר את קבוצת הביטויים שכבר שוערכו ואין צורך לשערכים שוב.

תארי את ריצת האלגוריתם בכל נקודה בתוכנית, ואת תוצאת ההרצה.

```
x := y
z := x + y
while (true) do:
  if(z == z * z) do:
    z := z - x
    y := y + x
z := x - 1
```