חורף 2017-18, מועד א' – שאלה 2

א. שלב ראשון: מריצים אנליזת Interval (בדיוק כפי שנלמד בהרצאה — אין צורך לפרט).
שלב שני: לכל משתנה x מתאימים את הטיפוס הראשון בטבלה (לפי הסדר מלמעלה למטה) שמכיל את הטווח של x עבור כל הנקודות בתכנית ש x מופיע בהן.

ב. אף אנליזה (ובפרט זו המוצעת) לא יכולה להיות מדויקת מכיוון שאי אפשר לוודא עצירה של תכניות שרירותיות. למשל בתכנית:

```
v := 1
while (…) do ( some code )
v := 1000
```

הטיפוס של v צריך להיות uint8 במקרה שהלולאה עוצרת, ו int16 אחרת, אבל לא ניתן לקבוע זאת.

ג. (יש מגוון תשובות אפשריות)
לא ניתן לבצע את האופטימיזציה באותו אופן בגלל קיומן של השמות בין מצביעים. למשל בתכנית:

```
a := new [n];
b := a;
a[0] := 1;
b[1] := 1000;
```

האנליזה שהצענו (אפילו אם מתעלמים מקיום האינדקסים) תתאים טיפוס int8 למערך a ו int16 למערך b. אבל אם נגדיר את הטיפוסים בצורה זו אז: (1) ההשמה a:=b לא תהיה חוקית יותר. (2) האופטימיזציה אינה sound מכיוון ש b,a מצביעים לאותו זכרון (aliasing), לפיכך ההשמה של 1000 תגלוש מהגודל של int8 שהוקצה ל a.

ד. מריצים במקביל לאנליזה של סעיף א גם אנליזת to-points (בדיוק כפי שנלמד — אין צורך לפרט).
בדומיין ה Intervals, בנוסף למשתנים, יהיה איבר אבסרקטי גם עבור כל אובייקט מוקצה דינמית, בהפרדה על פי allocation site. הסמנטיקה האבסטרקטית תהיה:

- עבור משתנים וביטויים מטיפוס מספרי: זהה לסעיף א (Intervals הרגיל).
- גישה לאיבר במערך מהצורה a[i]: הסמנטיקה היא join על-פני כל הטווחים ש a יכול להצביע אליהם באותו מצב (לפי יחס ה to-points).
- השמה לאיבר במערך מהצורה a[i]:= e: יש לעדכן את הערך האבסטרקטי של כל אובייקט ש a יכול להצביע עליו (לפי יחס ה to-points) באופן הבא:

```
s#'x = s#x join [[e]]s#    for all x in pts(a)
```