

פתרי 6 מתוך 8 השאלות הבאות

1. JavaScript and Closure

ברצוננו להצמיד לכל מפורסם מזהה ייחודי:

```
// This example is explained in detail below (just after this code
box).
function celebrityIDCreator (theCelebrities) {
  var i;
  var uniqueID = 100;
  for (i = 0; i < theCelebrities.length; i++) {
    theCelebrities[i]["id"] = function () {
      return uniqueID + i;
    }
  }
  return theCelebrities;
}

var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0},
{name:"Willis", id:0}];

var createIdForActionCelebs = celebrityIDCreator (actionCelebs);

var stalloneID = createIdForActionCelebs
[0]; console.log(stalloneID.id());
```

- א. מה הבאג ולמה הוא גורם (מה יודפס לקונסול ומה התכוונו שיודפס)?
- ב. תקני את הבאג תוך שימוש ב-Closure

2. חידת איינשטיין ופרולוג

נתון הדקדוק הבא:

```
Expr ::= + Expr Expr | * Expr Expr | Num | Xer
Xer   ::= x | ^ x Num
Num   ::= 2 | 3 | .... a Integer (bigger than 1) ...
```

א. רשמי פרידקט בפרולוג שיבדוק האם ביטוי הוא מהשפה שהוגדרה. מצורפות דוגמאות הרצה.

```
?- phrase(expr, [+ , * , 2 , x , ^ , x , 5 ] ).
true ;
false.
```

```
?- phrase(expr, [+ , * , * , 2 , x , ^ , x , 5 ] ).
false.
```

ב. מה יחזיר הביטוי הבא?

```
?- length(L, N), phrase(expr, L).
```

ג. הרחיבי את הכללים כך שיחזירו גם את עץ הגזירה. דוגמא הרצה מצורפת.

```
phrase(expr(AST), [+ , * , 2 , x , ^ , x , 5 ]), phrase(expr(AST), L).
```

3. Haskell Lazy Evaluation

נתבונן בקוד הבא:

```
quickSort [] = []
quickSort (x:xs) = quickSort (filter (< x) xs) ++ [x] ++ quickSort
(filter (>= x) xs)
```

א. הפעילי את הפונקציה על הקלט [3 2 1]

ב. התבונני בקוד הבא:

```
minimum ls = head (quickSort ls)
```

מה יקרה כשנקרא לו?

ג. מה היה קורה אם הביצוע לא היה עצל?

4. Types

נתון מימוש שגוי של הפונקציה append ב Haskell.

```
isPrefixOf [] _ = True
isPrefixOf _ [] = False
isPrefixOf (x:xs) (y:ys) = isPrefixOf xs ys
```

א. נתחי את הטיפוס של פונקצית isPrefixOf.

ב. זהו את הבאג בפונקציה ותקני אותו.

ג. נתחי את הטיפוס של הפונקציה המתוקנת.

5. Lua

השתמשי ב coroutines של Lua על מנת לכתוב שרשרת של יצרן-לקוח. הפונקציונאליות הנדרשת מהשרשרת היא קריאה של זרם מספרים אחד אחרי השני. כל מספר מייצג הצבעה למועמד כלשהו. יש להתעלם ממספרים שלילים או לא שלמים. יש לחשב את המנצח בבחירות. השרשרת צריכה להכיל לפחות 3 צמתים (2 מהם חייבים להיות ב coroutines). ניתן להניח קיומה של פונקציה הממיינת מספרים, אך יש לציין מה נדרש ממנה ואת חתימתה. (בנוסף: שימוש בזיכרון קבוע בהינתן שהמנצח מקבל יותר משליש מהקולות)

6. While שפת

הוחלט להוסיף לשפת While משפט repeat מהצורה repeat S until B; משמעות המשפט היא ביצוע של S לפחות פעם אחת עד אשר B מתקיים.

א. הרחיבי את הסמנטיקה הטבעית כדי לטפל במשפט.

ב. הדגימי את הסמנטיקה על התכנית הבאה:

```
z=3;repeat z=z*z; until z>80
```

ג. הרחיבי את סמנטיקת small step לטיפול במשפט הנ"ל.

ד. האם ניתן להגדיר עם הסמנטיקה הטבעית של repeat בעזרת סמנטיקת הטבעית של

While שנתנה בכיתה? אם כן, איך? אם לא, מדוע?

7. טיפוסים ב Haskell
נתבונן בקוד Haskell הבא:

```
data Tree = Leaf Int | Node (Int, Tree, Tree)
```

```
sum (Node(n,t1,t2)) = n + sum(t1) + sum(t2)
```

- א. תרגמי את הקוד לשפת C? האם הקוד עובר קומפילציה? האם הקוד נכון מבחינה לוגית? אם לא תקני אותו.
ב. האם הקומפיילר של Haskell יכול לגלות את הבעיה? ואיך? תקני את הקוד אם נדרש.

8. axiomatic semantics
התכנית הבאה מחשבת מחלק משותף גדול ביותר:

```
gcd ( int a, int b )  
{  
  int c;  
  while ( a != 0 ) {  
    c = a; a = b%a; b = c;  
  }  
  return b;  
}
```

- א. נסחי במדויק את תנאי הנכונות של השאלה.
ב. הוכיחי בעזרת אקסיומות Hoare שהתכנית נכונה.
ג. תני דוגמא לשינוי של מערכת האקסיומות שפוגע בנאותות מערכת הכללים.