

פתרי 6 מתוך 8 השאלות הבאות

1. JavaScript and Closure

ברצוננו להצמיד לכל תיבת טקסט אירוע לחיצה שמעלה הודעה עם מספרה הסידורי. במימוש הבא ישנו באג:

```
for (var i = 0; i < 10; i++) {  
  document.getElementById('box' + i).onclick = function() {  
    alert('You clicked on box #' + i);  
  };  
}
```

- א. מה הבאג ולמה הוא גורם?
- ב. תקני את הבאג תוך שימוש ב-Closure:

```
for (var i = 0; i < 10; i++) {  
  document.getElementById('box' + i).onclick =  
  
}
```

2. חידת איינשטיין ופרולוג

איינשטיין כתב את החידה וטען שרק 2% מכל העולם מסוגלים לפתור את החידה. ישנם 5 בתים בצבעים שונים. בכל בית גר דייר בעל לאום שונה. כל אחד מהדיירים שותה סוג משקה אחד בלבד, מעשן סוג סיגריות אחד בלבד ומגדל סוג אחד של בע"ח. אין דיירים שמחזיקים את אותה חיה או שותים את אותו משקה או מעשנים את אותו סוג סיגריות.

השאלה היא: מי מאנשים מחזיק דגים?

הנחות:

- הדייר הבריטי גר בבית האדום
- הדייר הדני שותה תה
- הדייר השוודי מגדל כלבים
- הבית הירוק נמצא משמאל לבית הלבן
- הדייר בבית הירוק שותה קפה
- הדייר שמעשן "פול-מל" מגדל ציפורים
- הדייר בבית הצהוב מעשן "דנהיל"
- הדייר בבית האמצעי שותה חלב
- הדייר הנורווגי גר בבית הבית הראשון
- הדייר שמעשן "בלנד" גר ליד הדייר שמגדל חתולים
- הדייר שמגדל סוסים גר ליד הדייר שמעשן "דנהיל"
- הדייר שמעשן "בלו-מסטר" שותה בירה

הדייר הגרמני מעשן "פרינס"
הדייר הנורווגי גר צמוד לבית הכחול
לדייר שמעשן "בלנד" יש דייר שכן ששותה מים

רשמי פרדיקטים בפרולוג שמתארים את כל אילוצי החידה אשר יאפשרו לפרולוג לפתור אותה ולהיות חלק מה- 2%. מותר להשתמש ב `fd_domain` וב- `fd_all_different`.

3. Haskell Lazy Evaluation
נתבונן בקוד הבא:

```
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

א. חשבי את `take 3 fibs`
ב. הסבירי מה היה קורה אם הארגומנטים ל `fibs` היו מחושבים בצורה ישירה (`eager`) כמו בשפות Java או C.
ג. מהן היתרונות של העובדה ששפת Haskell היא עצלנית (`Lazy`) בהגדרת מבנים חדשים?

4. Types
נתון מימוש שגוי של הפונקציה `append` ב Haskell.

```
append([],r)=r  
append(x:xs,r)=append(xs,r)
```

א. נתחי את הטיפוס של פונקצית `append`.
ב. זהו את הבאג בפונקציה ותקן אותו.
ג. נתחי את הטיפוס של הפונקציה המתוקנת.

5. Lua
השתמש ב `coroutines` של Lua על מנת לכתוב שרשרת של יצרן-לקוח. הפונקציונאליות הנדרשת מהשרשרת היא קריאה של זרם מספרים אחד אחרי השני וחישוב של ממוצע, מקסימום, מינימום וחציון של טווח המוגדר מראש ולא ניתן לשינוי תוך כדי הריצה. השרשרת צריכה להכיל לפחות 3 צמתים (2 מהם חייבים להיות ב `coroutines`). ניתן להניח קיומה של פו' מיון הממיינת מספרים, אך יש לציין מה נדרש ממנה ואת חתימתה.

6. שפת While
הוחלט להוסיף לשפת While משפט `Random` מהצורה `ID:=random;` משמעות המשפט היא ערך רנדומי מטיפוס `Integer` למשתנה `ID`.
א. הרחיבי את הסמנטיקה הטבעית כדי לטפל במשפט.
ב. הדגמי את הסמנטיקה על התכנית הבאה:

```
x=random;  
if x>7 then y=5;  
else y=10;
```

ג. הרחיבי את סמנטיקת `small step` לטיפול במשפט הנ"ל.
ד. האם ניתן להגדיר עם הסמנטיקה הטבעית `random` בעזרת סמנטיקת `ora` שנתנה בכיתה? אם כן, איך? אם לא, מדוע?

7. טיפוסים ב Haskell

נתבונן בקוד C הבא המשתמש ב union כדי לשמור איחוד בין טיפוסים:

```
union {
    int i;
    float f;
} int_or_float;

float ff;
int flag;
if(cond){
    flag=1;
    int_or_float.i=5;
}
else {
    flag=0;
    int_or_float.f=5.5;
}
...
if(flag)
    ff= int_or_float.f;
else
    ff=(float) int_or_float.i;
```

- הניחו של-`cond` יש לעתים ערך 0 ולעתים לא. טיפוס אלגברי ב-Haskell הוא איחוד של טיפוסים עם סימן |.
- הסבירי מה יקרה כשקוד זה ירוץ?
 - תקני את הקוד.
 - האם ניתן לכתוב בשפת Haskell קוד שקול לקוד המתוקן תוך שימוש בטיפוסים אלגבריים? האם ניתן לתרגם את הקוד המקורי ל Haskell?

8. axiomatic semantics

התכנית הבאה מבצעת כפל מספרים שלמים:

```
mul(m,n)
{
    ret=0;
    i=n;
    while(i>0)
    {
        ret+=m;
        i--;
    }
    return ret;
}
```

א. נסחי במדויק את תנאי הנכונות של השאלה.

- ב. הוכחי בעזרת אקסיומות Hoare שהתכנית נכונה.
- ג. תני דוגמא לשינוי של מערכת האקסיומות שפוגע בשלמות מערכת הכללים.