

עבודה להגשה מס' 4

- הגשת עבודה בכתב קריא בקובץ PDF אחד בלבד דרך המודל.
- (אין לשלוח עבודה במייל!)
- כל יום איחור בהגשת עבודה מוריד 5 נקודות מהציון.
- אין קבלת עבודות באיחור של יותר מ-3 ימים.

דקדוק עם פעולות סמנטיות

עבור הדקדוק והפעולות הסמנטיות שלו יחד עם הקלט הנתון, ענו על השאלות מטה:

Shape Description Language

Production	Semantic Rules
Program -> Statement Program	Program.shapes = Statement.shape + Program.shapes
Program -> Statement	Program.shapes = [Statement.shape]
Statement -> Shape	Statement.shape = Shape.obj
Statement -> Transform	Statement.shape = Transform.obj
Shape -> CIRCLE radius	Shape.obj = Circle(radius.value)
Shape -> SQUARE side	Shape.obj = Square(side.value)
Transform -> MOVE shape dx dy	Transform.obj = Move (shape.obj, dx.value, dy.value)
Transform -> ROTATE shape angle	Transform.obj = Rotate(shape.obj,angle.value)

והקלט לפי הסדר:

CIRCLE 5
 SQUARE 3
 MOVE CIRCLE 2 3
 ROTATE SQUARE 45

1. מהו הפלט שמתקבל?
2. מהן התכונות הנורשות (inherited) והנוצרות (synthesized) (אם קיימים) ? הסבירו.
3. בנו גרף תלויות עבור כל הקלט
4. על סמך גרף תלויות מהסעיף הקודם, קבעו את סדר הפעולות הסמנטיות ובצעו אותן.

3AC ואופטימיזציות

1. נתונה פונקציה הכתובה בשפת C,
 - א. תרגמו את הקוד לפונקציה הכתובה ב-TAC (3 address code).
 - א.1: ה-LABELS חייבים להיות ממוינים בסדר עולה.
 - א.2: יש לטפל בביטוי לוגי מורכב לפי עיקרון של-short circuit evaluation.
 - ב. יש לבצע אופטימיזציה ל-TAC שהמרתם. בין היתר, צמצמו זמן ריצה של הפונקציה ככל שניתן ללא פגיעה בקוד (לדוגמא – הוצאת קוד מלולאות).

```
void fn(int a, int b, int c, int d){
    double arr[30] = {0.0};
    int i;
    for (i = 0; i < d*1000; ++i) {
        if ( (c % 7==0) && (a / b) >1 ){
            arr[i] = ( (c % d) * a / b) % d) * i;
            a -=1;
        }
        if (b<a && b<c)
            printf ("arr[%d]=%d",i,arr[i]+sqrt (b)*pow(2,d));
    }
}
```

המשך בעמוד הבא....

נתונה הפונקציה הבאה הכתובה ב-3AC:

Func:

```

BeginFunc 68;
  _tmp0 = 1;
  _tmp1 = base < _tmp0;
  _tmp2 = base == _tmp0;
  _tmp3 = _tmp1 || _tmp2;
  IfZ _tmp3 Goto _L0;
  result = base;
  Goto _L1;
_L0:
  _tmp4 = 0;
  f0 = _tmp4;
  _tmp5 = 1;
  f1 = _tmp5;
  _tmp6 = 2;
  i = _tmp6;
_L2:
  _tmp7 = i < base;
  _tmp8 = i == base;
  _tmp9 = _tmp7 || _tmp8;
  IfZ _tmp9 Goto _L3;
  _tmp10 = f0 + f1;
  result = _tmp10;
  f0 = f1;
  f1 = result;
  _tmp11 = 1;
  _tmp12 = i + _tmp11;
  i = _tmp12;
  Goto _L2;
_L3:
_L1:
  Return result;
EndFunc;

```

- א. יש לבצע אופטימיזציה של צמצום משתנים (רגיסטרים) בתוך הפונקציה הנתונה.
- ב. ציירו את ה- interference graph
- ג. צבעו את הגרף במינימום צבעים לפי אלגוריתם הצביעה האוריסטית. אם יש יותר מקודקוד אחד מדרגה מינימלית, יש להוריד קודם את הקודקוד עם הערך הקטן ביותר. (למשל אם יש קודקודים b,f,z בעלי אותה דרגה מינימלית, יש להוריד את b ראשון).

שאלות בנוגע לעבודה יש להפנות למתרגל בלבד aaccbb@gmail.com או במסר במודל

בהצלחה