

צ'ודנובסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 14

שאלה 1: נתון דקדוק G נרחיב אותו (הרחבה קשורה רק לסעיף ב') ונקבל:

- (1) $S' \rightarrow S$
- (2) $S \rightarrow ScAB$
- (3) $S \rightarrow bSc$
- (4) $S \rightarrow a$
- (5) $A \rightarrow aA$
- (6) $A \rightarrow a$
- (7) $B \rightarrow Bb$
- (8) $B \rightarrow \epsilon$

א. נוסיף פעולות סמנטיות כך שבמהלך קריאת הקלט, כל סימן a יודפס פעמיים:

- (1) $S \rightarrow ScAB$
- (2) $S \rightarrow bSc$
- (3) $S \rightarrow a \{ \text{print}("aa"); \}$
- (4) $A \rightarrow a \{ \text{print}("aa"); \} A$
- (5) $A \rightarrow a \{ \text{print}("aa"); \}$
- (6) $B \rightarrow Bb$
- (7) $B \rightarrow \epsilon$

ב. נשתמש בתכונות מורשות הבאות:

- $S.pre_count$ - מספר האותיות שמופעים בקלט לפני S (כלומר לפני האותיות הנגזרים מ- S).
- $A.pre_count$ - מספר האותיות שמופעים בקלט לפני A (כלומר לפני האותיות הנגזרים מ- A).
- $S.preLetterIs_b$ - אות אחרונה שמופיע בקלט לפני S (כלומר לפני האותיות הנגזרים מ- S).
- $A.preLetterIs_b$ - אות אחרונה שמופיע בקלט לפני A (כלומר לפני האותיות הנגזרים מ- A).

נשתמש גם בתכונות הנבנות הבאות:

- $S.count$ - מספר אותיות הקלט הנגזרות מ- S .
- $A.count$ - מספר אותיות הקלט הנגזרות מ- A .
- $B.count$ - מספר אותיות הקלט הנגזרות מ- B .
- $S.string$ - מחרוזת של אותיות a שנאספת במהלך התהליך ומדפיסים אותה עם עונה על תנאים.
- $A.string$ - מחרוזת של אותיות a שנאספת במהלך התהליך.

צ'ודנובסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 14

```
S' -> { S.pre_count = 0; S.preLetterls_b = false; } S { if ( S.count > 9 ) printf ( S.string ); }

S -> { S1.pre_count = S.pre_count; S1.preLetterls_b = S.preLetterls_b; } S1
      c { A.pre_count = S1.pre_count + S1.count + 1; A.preLetterls_b = false; } A B
      { S.count = S1.count + A.count + B.count + 1; S.string = S1.string + A.string; }

S -> b { S1.pre_count = S.pre_count + 1; S1.preLetterls_b = true; } S1
      c { S.count = S1.count + 2; S.string = S1.string; }

S -> a { S.count = 1; S.string = " ";
        if ( ( S.pre_count < 8 ) && ( S.preLetterls_b == false ) ) S.string += "a"; }

A -> a { A1.pre_count = A.pre_count + 1; A1.preLetterls_b = false; } A1
      { A.count = A1.count + 1; A.string = A1.string;
        if ( ( A.pre_count < 8 ) && ( A.preLetterls_b == false ) ) A.string += "a"; }

A -> a { A.count = 1; A.string = " ";
        if ( ( A.pre_count < 8 ) && ( A.preLetterls_b == false ) ) A.string += "a"; }

B -> B1 b { B.count = B1.count + 1; }

B -> epsilon { B.count = 0; }
```

צ'ודנבסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 14

שאלה 2:

```
S -> tree {
    if(tree.value > 0) print ( tree.value );
    else print ( "error" );
}

tree -> SUM_ODD '(' treelist ')' {
    tree.value = sum_odd(treelist.list);
    tree.size = treelist.nodeCount;
}

tree -> SUM_EVEN '(' treelist ')' {
    tree.value = sum_even(treelist.list);
    tree.size = treelist.nodeCount;
}

tree -> SIZE '(' treelist ')' { tree.size = treelist.nodeCount + 1 ; tree.value = tree.size ; }

tree -> IGNORE '(' treelist ')' { tree.size = 0 ; tree.value = 0 ; }

tree -> NUMBER { tree.size = 1 ; tree.value = NUMBER.val ; }

treelist -> treelist1 tree {
    treelist.nodeCount = treelist1.nodeCount + tree.size + 1;
    treelist.list = add(treelist1.list , tree.value);
}

treelist -> tree {
    treelist.nodeCount = tree.size + 1 ;
    treelist.sum = makelist (tree.value);
}
```

tree.value - ערך של העץ (תכונה נבנית).

tree.size - סופרת כמות צמתים בעץ (תכונה נבנית).

treelist.list - רשימת צמתים שזה בנים של הצומת (תכונה נבנית).

treelist.nodeCount - סופרת כמות צמתים ברשימה (תכונה נבנית).

sum_odd - פונקציה מחשבת סכום ברשימה של איברים אי-זוגיים.

sum_even - פונקציה מחשבת סכום ברשימה של איברים זוגיים .

add - פונקציה מוסיפה לרשימה איבר חדש.

makelist - פונקציה יוצרת רשימה עם איבר אחד .

צ'ודנובסקי דמיטרי 324793900 קומפילציה 20364 א 2021 ממ"ן 14

שאלה 4:

א. נתונה קטע קוד הבא:

```
for(j = 0; j < 10; j++) if(a > 10) bar = 2 * bar;
```

נכתוב את משפט הזה בשפת QUAD:

- (1) IASN j 0
- (2) ILSS t1 j 10
- (3) JMPZ 10 t1
- (4) IGRT t2 a 10
- (5) JMPZ 8 t2
- (6) IMLT t3 2 bar
- (7) IASN bar t3
- (8) IADD j j 1
- (9) JUMP 2
- (10) HALT

ב. נתונה קטע קוד הבא:

```
switch(10 * k - 1){  
    case 19: a = a + 7; break;  
    case 99: a = a - 4; break;  
    default: a = 0;  
}
```

נכתוב את משפט הזה בשפת QUAD:

- (1) IMLT t1 10 k
- (2) ISUB t2 t1 1
- (3) IEQL t3 t2 19
- (4) JMPZ 8 t3
- (5) IADD t4 a 7
- (6) IASN a t4
- (7) JUMP 14
- (8) IEQL t3 t2 99
- (9) JMPZ 13 t3
- (10) ISUB t5 a 4
- (11) IASN a t5
- (12) JUMP 14
- (13) IASN a 0
- (14) HALT