

Programski prevodioci: Projekat

Šifra:PP-FdsC

Sadržaj

1. Napomena	1
2. Zadatak 1	1
3. Zadatak 2	1
4. Zadatak 3	2

1. Napomena

Za svaki zadatak potrebno je uraditi:

1. Sintaksnu analizu
2. Semantičku analizu
3. Generisanje koda

2. Zadatak 1

Omogućiti višestruku dodelu vrednosti.

```
int a, b = 100
unsigned int c,d = 3u
```

3. Zadatak 2

Proširiti jezik PARA iskazom koji ima sledeći oblik:

```
"para" "(" <id> "=" <lit1> ":" <lit2> ":" "paso" <lit3> ")"
<statement>
```

Gde:

- <id> predstavlja lokalnu promenljivu ili parametar
- <lit1> predstavlja literal, vrednost na koju se inicijalizuje iterator
- <lit2> predstavlja literal, maksimalna vrednost koju iterator može da dostigne
- <lit3> predstavlja literal, korak pomeranja iteratora
- <statement> predstavlja iskaz

Realizovati semantičke provere:

1. <id> mora biti prethodno deklarisan.
2. <id>, <lit1>, <lit2> i <lit3> moraju biti istog tipa.
3. Vrednost <lit1> mora biti manja od vrednosti <lit2>

Izvršavanje:

- Inicijalizacija iteratora se vrši samo jednom, pre prvog izvršavanja petlje (<lit1>).
- Provera vrednosti iteratora se vrši na početku svake iteracije. Telo petlje se izvršava ako je iterator striktno manji od <lit2>.
- Nakon svake iteracije, vrednost iteratora se uvećava za vrednost <lit3>.



Omogućiti i ugnježdene para iskaze.

Primer:

```
int zbir = 0;
int razlika = 0;
para ( i = 0 : 5 : paso 2){
    zbir = zbir + i;
    razlika = razlika - i;
}
para ( i = 1 : 13 : paso 4)
    razlika = zbir - i;
```

4. Zadatak 3

Proširiti jezik CHECK iskazom koji ima sledeći oblik:

```
"check" "[" <check_expression> "]" "{"
    "when" <constant_expression> "->" <when_body> ["finish" ";"]
    ...
    ["otherwise" "->" <otherwise_statement>]
"}"
```

Gde:

- <check_expression> predstavlja ime promenljive
- <constant_expression> predstavlja konstantu
- <when_body> predstavlja iskaz (statement)
- <otherwise_statement> predstavlja iskaz (statement)



Mora postojati bar jedna when naredba. Finish naredba se opciono može pojaviti samo na kraju when naredbe. Otherwise naredba je opciona i može se pojaviti samo posle svih when naredbi

Realizovati sledeće semantičke provere:

1. Promenljiva u <check_expression> mora biti prethodno deklarirana
2. Konstante u svim when iskazima moraju biti jedinstvene
3. Tip konstante u when naredbi mora biti isti kao tip promenljive u <check_expression>

Izvršavanje:

- Na početku check iskaza se izvrši provera vrednosti promenljive check_expression.
- U zavisnosti od te vrednosti preusmerava se tok izvršavanja na telo odgovarajuće when naredbe.
- Ukoliko se na kraju when naredbe nalazi finish naredba, tok izvršavanja se preusmerava na kraj check iskaza; a ako je finish naredba izostavljena, "propada" se na izvršavanje sledeće when naredbe.
- Otherwise naredba se izvršava ukoliko se vrednost check promenljive razlikuje od svih konstanti navedenih u svim when naredbama.

Primer:

```
check [a]{  
  when 1 ->  
    a = a + 5;  
    finish;  
  when 5 ->  
  {  
    b = 3;  
  }  
  otherwise ->  
    a = a + b;  
}
```