

Programski prevodioci: Vežbe 9

Sadržaj

1. Uvod	1
2. Zadaci	1
2.1. Zadatak 1: switch iskaz	1
2.2. Zadatak 2: iterate	2
2.3. Zadatak 3: branch iskaz	3

1. Uvod

U ovoj nedelji biće rađeni zadaci vezani za generisanja koda.

2. Zadaci

2.1. Zadatak 1: **switch** iskaz

Proširiti miniC gramatiku **switch** iskazom.

Sintaksa **switch** iskaza ima oblik:

```
"switch" "(" switch_expression ")" "{"  
  "case" constant_expression ":" case_body [ "break" ";" ]  
  "case" constant_expression ":" case_body [ "break" ";" ]  
  ...  
  [ "default" ":" default_body ]  
"}"
```

- **switch_expression** predstavlja ime promenljive koja prethodno mora biti deklarirana.
- **constant_expression** predstavlja konstantu.
- **case_body** i **default_body** predstavljaju iskaz. Postoji bar jedna **case** naredba.
- **default** naredba se može pojaviti samo nakon **case** naredbi (kao poslednja).
- **break** naredba se može pojaviti samo na kraju **case** naredbe.

Realizovati sledeće semantičke provere:

1. Promenljiva u **switch_expression** mora biti prethodno deklarirana.
2. Tip konstante u **case** naredbi mora biti isti kao tip promenljive u **switch_expression**.
3. Konstante u svim **case** iskazima moraju biti jedinstvene.

Izvršavanje:

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

Primer 1:

```
switch (state) {  
  case 10: { s = 1; } break;  
  case 20: s = 2;  
  default: s = 0;  
}
```

Primer 2:

```
switch (state) {  
  case 10: s = 1; break;  
  case 20: { s = 2; }  
}
```

2.2. Zadatak 2: **iterate**

Proširiti miniC iskaze iterator iskazom koji ima sledeći oblik:

```
"iterate" <name> <lit1> "to" <lit2> <statement>
```

Gde:

- <name> iterator, predstavlja ime lokalne promenljive ili parametra
- <lit1> literal koji predstavlja korak
- <lit2> literal koji predstavlja kraj iteracije
- <statement> predstavlja iskaze

Izvršavanje:

- Pre početka petlje treba postaviti iterator (name) na vrednost 1.
- Tačnost relacije se proverava na početku svake iteracije i izvršava se dokle god je <name> manje ili jednako <lit2>
- Nakon svake iteracije, vrednost iteratora se uvećava za korak lit1.



Omogućiti i ugnježdene iterate iskaze.

Primer:

```
int x;  
int y;  
y=0;  
iterate x 3 to 20 {  
    y = x + y;  
}
```

2.3. Zadatak 3: **branch** iskaz

Proširiti jezik BRANCH iskazom koji ima sledeći oblik:

```
"branch" "(" <var> ";" <const1> "," <const2> "," <const3> ")"  
    "first" <statement1>  
    "second" <statement2>  
    "third" <statement3>  
    "otherwise" <statement4>
```

Gde:

- <var> predstavlja ime promenljive
- <const1>, <const2> i <const3> predstavljaju konstante
- <statement1>, <statement2>, <statement3> i <statement4> predstavljaju iskaze



Uvek mora postojati tačno tri konstante i tačno četiri iskaza

Realizovati sledeće semantičke provere:

1. Promenljiva var mora biti prethodno deklarisana.
2. Konstante const1, const2 i const3 moraju biti istog tipa kao i var.

Izvršavanje:

- Na početku branch iskaza se izvrši provera vrednosti promenljive var.
- U zavisnosti od te vrednosti, ukoliko je ona jednaka vrednosti konstante const1 izvršava se first naredba, ukoliko je ona jednaka vrednosti konstante const2 izvršava se second naredba, ukoliko je ona jednaka vrednosti konstante const3 izvršava se third naredba.
- Nakon izvršavanja first, second ili third naredbe, tok izvršavanja se preusmerava na kraj branch iskaza.
- Otherwise naredba se izvršava ukoliko se vrednost promenljive var razlikuje od konstanti const1, const2 i const3.

Primer:

```
branch ( a ; 1 , 3 , 5 )  
  first a = a + 1;  
  second a = a + 3;  
  third a = a + 5;  
  otherwise a = a - 3;
```