

# Programski prevodioci: Vežbe 4

## Sadržaj

1. Uvod.....	1
2. Rešenja zadataka .....	1
2.1. Zadatak 1.....	1
2.2. Zadatak 2.....	2
2.3. Zadatak 3.....	2
2.4. Zadatak 4.....	3

## 1. Uvod

U dokumentu su data rešenja zadataka koji su rađeni na četvrtim vežbama.

## 2. Rešenja zadataka

Svi zadaci se rešavaju sledećim redosledom:

- Dodati nove tokene na vrh `.y` datoteke.
- Definirati regularne izraze u `.l` datoteci za nove tokene.
- Proširiti gramatiku jezika tako da sintaksno podržava novu konstrukciju.
- Dodati semantičke provere.

### 2.1. Zadatak 1

```
int type = -1;
%token _COMMA

variable
: _TYPE { type = $1; } vars _SEMICOLON ②
;

vars
: _ID
{
    if(lookup_symbol($1, VAR|PAR) != -1)
        err("redefinition of '%s'", $1);
    else
        insert_symbol($1, VAR, type, ++var_num, NO_ATR); ①
}
```

```

| vars _COMMA _ID
{
    if(lookup_symbol($3, VAR|PAR) != -1)
        err("redefinition of '%s'", $3);
    else
        insert_symbol($3, VAR, type, ++var_num, NO_ATR);
}
;

```

- ① Prilikom dodavanja novog simbola u tabelu, u treću kolonu je potrebno navesti tip simbola. Tip simbola se dobija kao semantička vrednost tokena `_TYPE`. Međutim, ovaj token se nalazi u okviru skroz drugog pravila, pa mu nije moguće pristupiti preko "dolar" promenljivih. Iz tog razloga, uvodi se globalna promenljiva `type`.
- ② Nakon tokena `_TYPE`, izvršava se akcija u kojoj semantičku vrednost tokena `_TYPE` čuvamo u globalnoj promenljivi `type`.

## 2.2. Zadatak 2

```

%token _INC

exp
:
| _ID _INC
{
    if(($$ = lookup_symbol($1, (VAR|PAR))) == -1)
        err("'%' undeclared", $1);
}
;

```

## 2.3. Zadatak 3

```

%token _DO
%token _WHILE

do_while_statement
: _DO statement _WHILE _LPAREN _ID _RELOP literal _RPAREN _SEMICOLON
{
    int idx = -1;
    if((idx = lookup_symbol($5, (VAR|PAR))) == -1)
        err("invalid identifier in do-while");
    if(get_type(idx) != get_type($7))
        err("incompatible types in do_while");
}
;

```

```

statement
: compound_statement
| assignment_statement
| if_statement
| return_statement
| do_while_statement ①
;

```

① Ukoliko se novonapravljeni pojam `do_while_statement` ne ubaci u `statement`, Bison će prijaviti upozorenje: `1 nonterminal useless in grammar`. Ovo upozorenje znači da postoji pojam u `.y` datoteci koji stoji sam za sebe i nema nikakve veze sa ostatkom gramatike, što znači da je njegovo postojanje beskorisno.

## 2.4. Zadatak 4

```

%type <i> num_exp exp literal function_call argument rel_exp ids

assignment_statement
: ids _ASSIGN num_exp
{
    if (get_type($1) != get_type($3))
        err("incompatible types");
}
_SEMICOLON
;

ids
: _ID
{
    $$ = lookup_symbol($1, VAR|PAR);
    if ($$ == -1)
        err("'s' undeclared", $1);
}
| ids _ASSIGN _ID
{
    int idx = lookup_symbol($3, VAR|PAR);
    if (idx == -1)
        err("'s' undeclared", $3);
    if (get_type($1) != get_type(idx))
        err("incompatible types");

    $$ = $1;
}
;

```