

Type Checking Assignment

(Individual submission)

Submission Deadline: 26/10/2019

Generate grammar for arithmetic expression using +, -, *, and / operators, assignment statement and declaration for int, char, 1 dimensional array of integer, non-nested struct.

Write type a checking system (in bison) that will check the type compatibility of the operators for programs written in this grammar.

If the types of operands of an arithmetic operator are different, implicit type conversion is done or explicit type conversion is required. In the first case, three address code is generated and a message is displayed.

Examples of inputs and outputs are as follows.

Input:

```
int a, b;  
float c, d;  
b = c + d;
```

Expected output:

No three address code and the following message:

“Type casting (Explicit type conversion) in the assignment operator in line number 3 is required.”

Input:

```
int a, b;  
float c, d;  
d = b + c * a;
```

Expected output:

Normal Three address code and the following message

“Type coercing (Implicit type conversion) from int to float is done in ‘+’ operator; Type coercing (Implicit type conversion) from int to float is done in ‘*’ operator.”

Input:

```
struct s {  
    int x, y;  
}  
Int a, b;  
s c;  
c = a;
```

Expected output:

The following error message

“Type conversion from int to struct cannot be done.”

Input:

```
int a, b;  
int c[10];  
b = c;
```

Expected output:

The following error message

“Type conversion from array to int cannot be done”