PES UNIVERSITY ONLINE

Compiler Design

Preet Kanwal

Department of Computer Science & Engineering

Compiler Design

Building a Mini Compiler - Intermediate Code Generation

Preet Kanwal

Department of Computer Science & Engineering

Compiler Design

Implementation tasks

- Given an expression, generate Intermediate code.
- The generated code must be in the quadruple form.

Compiler Design

Expected Results

Sample Input:

```
x = 9 / 2 + a - b
```

Output

```
/, 9, 2, t1
+, t1, a, t2
-, t2, b, t3
=, t3, , x
```

Compiler Design

Expected Results

Sample Input:

```
b = c / 6.7 + 12.45 - a * 1234.0
```

Output

```
/, c, 6.7, t1
+, t1, 12.45, t2
*, a, 1234.0, t3
-, t2, t3, t4
=, t4,  , b
```

Intermediate code generation implementation: yacc file

- Intermediate code needs to be generated only for a valid expression.

```
ASSGN : T_ID '=' E  {   //call quad_code_gen with appropriate parameters    }
    ;

E : E '+' T     {
                    /*create a new temporary and call quad_code_gen with appropriate
                    parameters*/
                }
    | E '-' T   {
                    /*create a new temporary and call quad_code_gen with appropriate
                    parameters*/
                }
    | T
    ;
```

```
T : T '*' F     {
                    /*create a new temporary and call quad_code_gen with appropriate
                    parameters*/
                }
    | T '/' F   {
                    /*create a new temporary and call quad_code_gen with appropriate
                    parameters*/
                }
    | F
    ;

F : '(' E ')'   {   //assign the value of node E to node F  }
    | T_ID      {   //assign a copy of t_ID of node F   }
    | T_NUM     {   //assign a copy of t_ID of node F   }
    ;
```

Intermediate Code Generation implementation: yacc file

- Let's take an example expression:
  a = 45 + 24;

- Let the grammar be:

```
ASSGN : T_ID '=' E
E : E '+' T | T
T : T_NUM
```

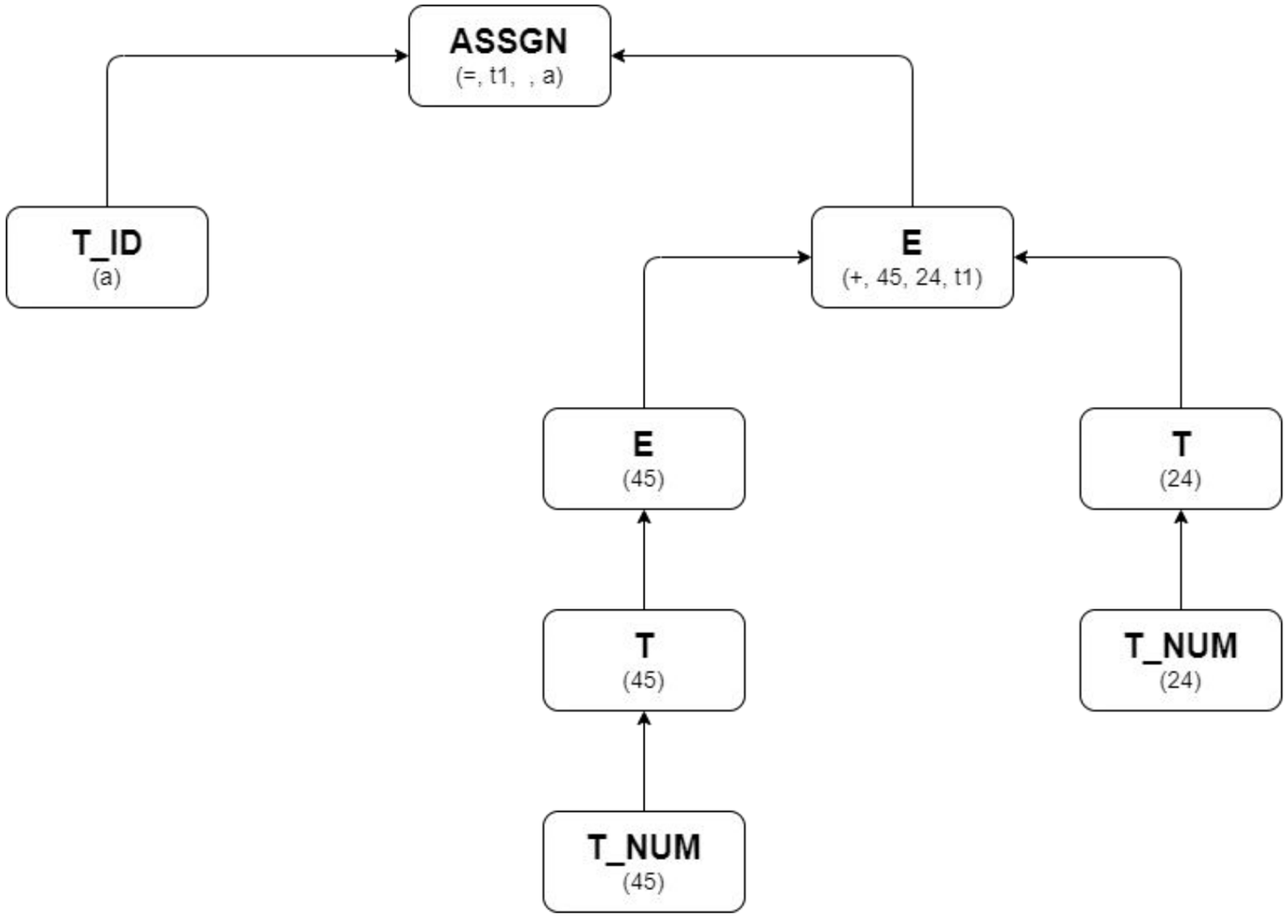
PES UNIVERSITY ONLINE

Important points:

- You will have to complete the helper functions to print the intermediate code.
- The generated code must be written to the file initialised in the main function of parser.y (use fprintf to write to the file)

Compiler Design

Mini-Compiler



So our grammar would look like this

```
ASSGN : T_ID '=' E  {
                        generate_icg($$, $3, "=", " ");
                    }
E : E '+' T {
                $$ = new_temporary();
                generate_icg($$, $1, "+", $3);
            }
    | T
T : T_NUM
```

THANK YOU

Preet Kanwal

Department of Computer Science & Engineering

preetkanwal@pes.edu