Compiler Design Laboratory (CS 753)

Samit Biswas

samit@cs.iiests.ac.in

Department of Computer Science and Technology,
Indian Institute of Engineering Science and Technology, Shibpur

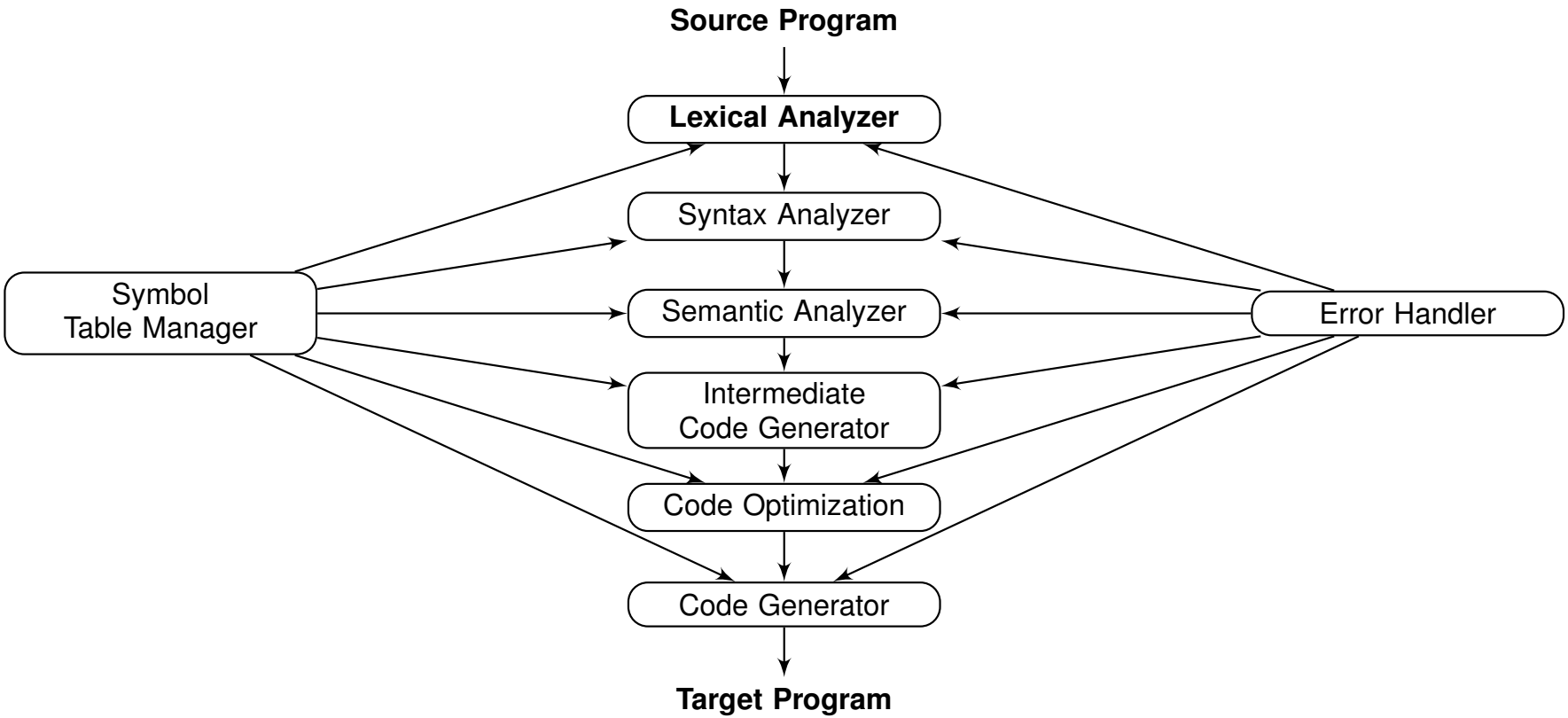
July 30, 2019

Phases Of Compilation

Lexical Analyzer

Assignment

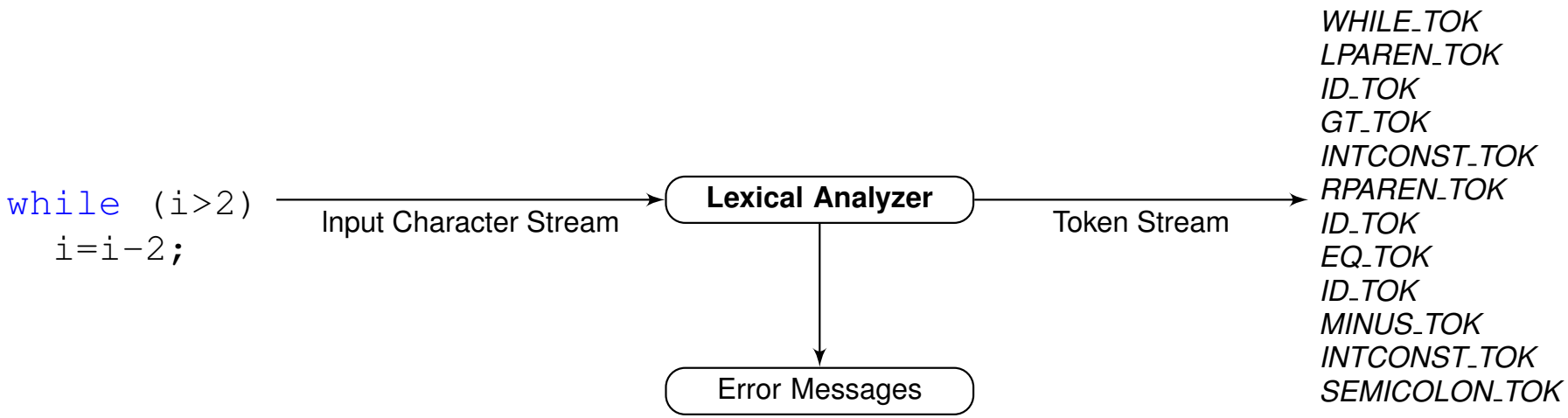
Phases Of Compilation



Lexical Analyzer

- converts the input program into a sequence of Tokens.
- can be implemented with the help of *Finite Automata*.

Lexical Analyzer



Programmer's View

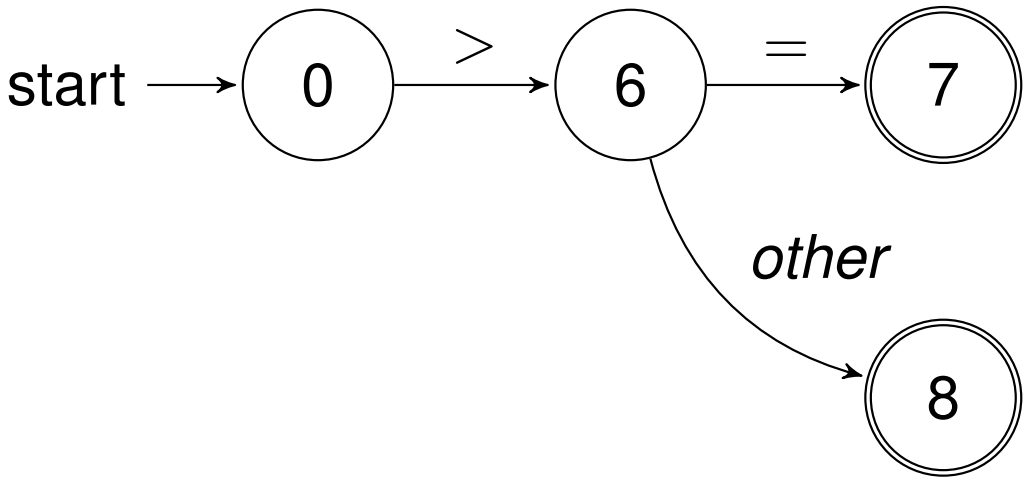
```
FILE *yyin;
char *yytext;
main(int argc, char *argv[]){
int token;
if (argc != 2){

}else{
        yyin = fopen(argv[1], "r");
        while(!feof(yyin)){
                token = yylex();
                printf("%d", token);
        }
        fclose(yyin);
    }
}
```

```
int yylex(){
    ...
    ...
}
```

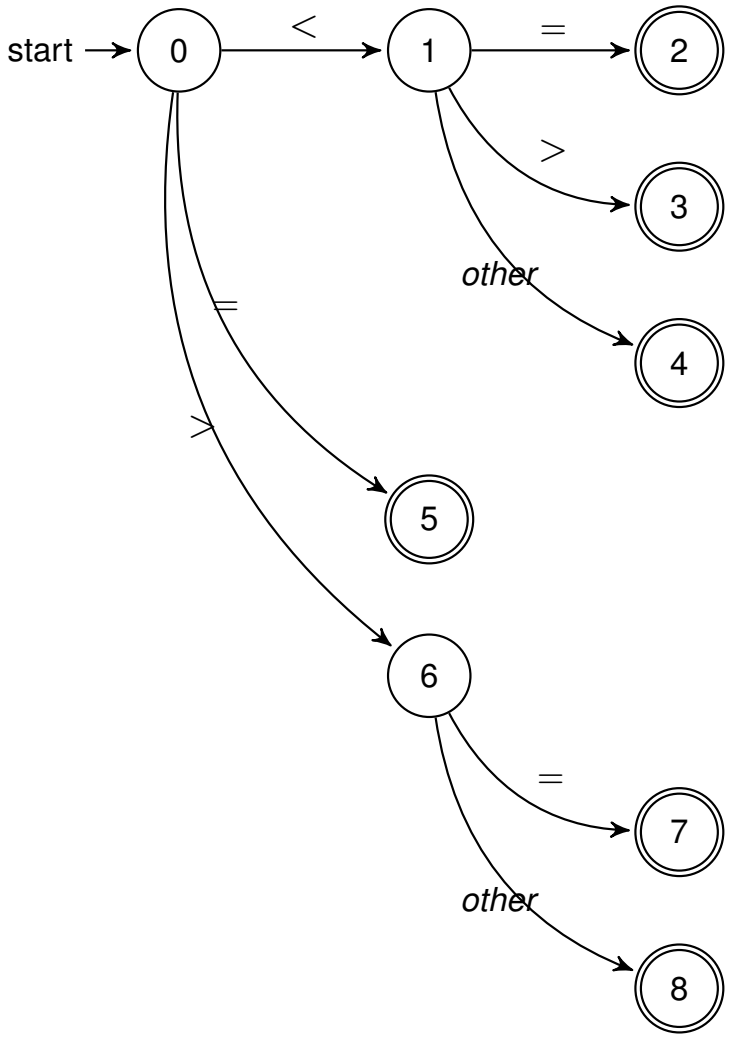
Loop and switch Approach

```
/* Single caharacter lexemes */
#define LPAREN_TOK '('
#define GT_TOK '>'
#define RPAREN_TOK ')'
#define EQ_TOK '='
#define MINUS_TOK '-'
#define SEMICOLON_TOK ';'
/*.....................
.....................*/
/* Reserved words */
#define WHILE_TOK 256
/*.....................
.....................*/
/* Identifier, constants..*/
#define ID_TOK 350
#define INTCONST 351
/*.....................
.....................*/
```

Based on the Concept of Deterministic finite Automata
Transition Diagram for $>=$

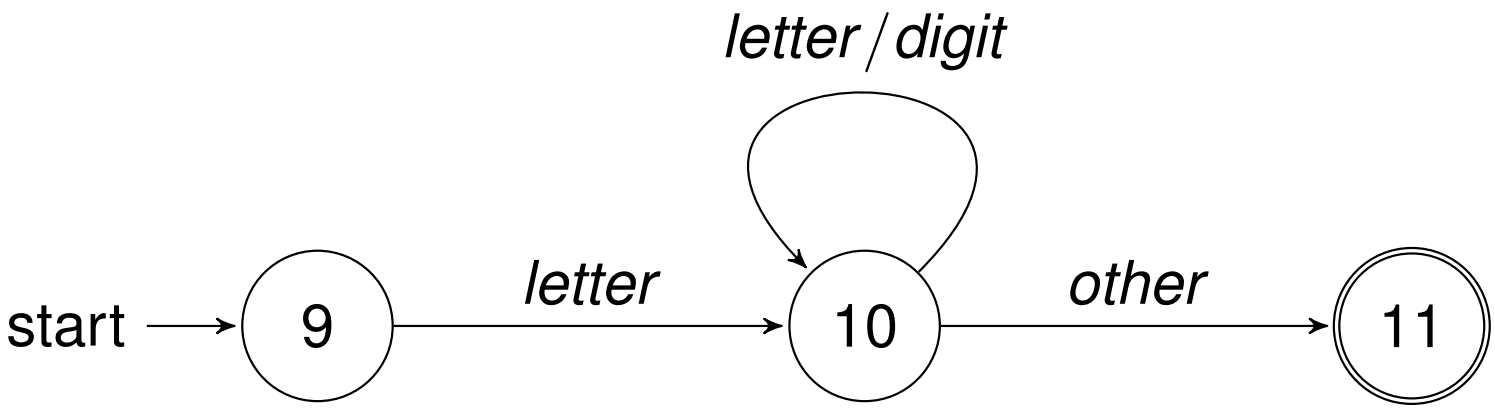


Transition Diagrams for Relational Operators



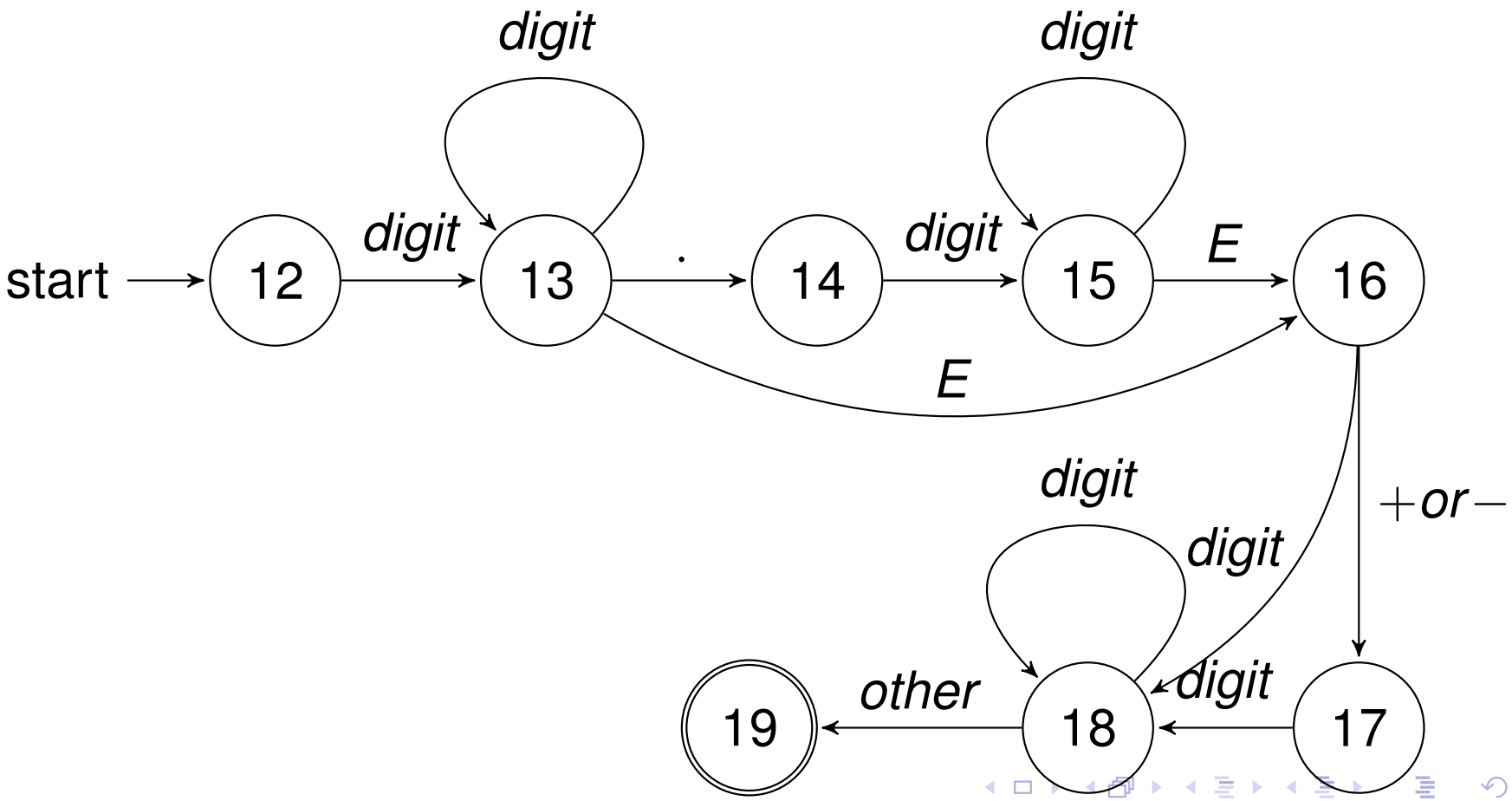
Recognitions of Tokens

Transition Diagrams for Identifiers or Keywords



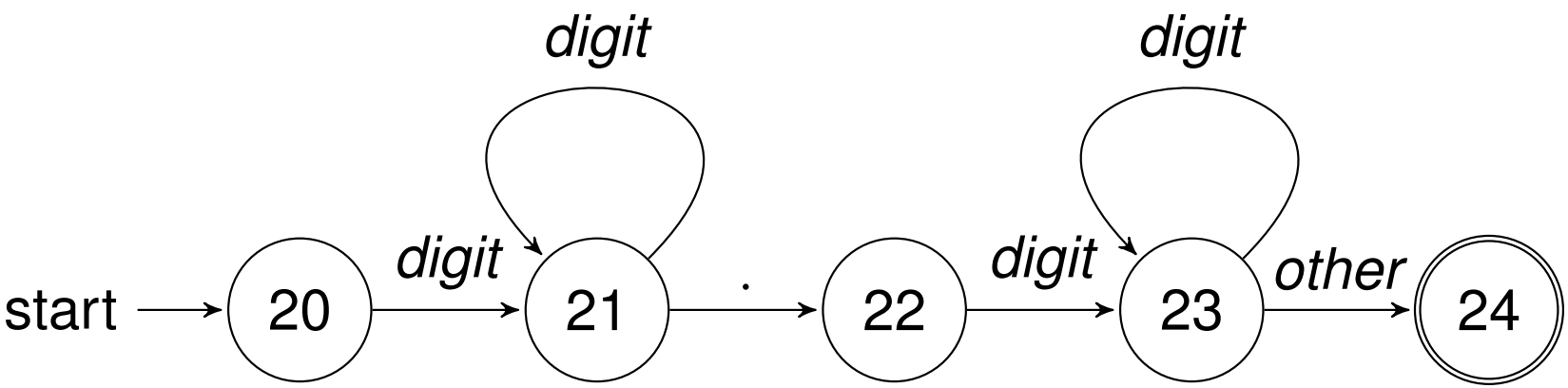
Recognitions of Tokens

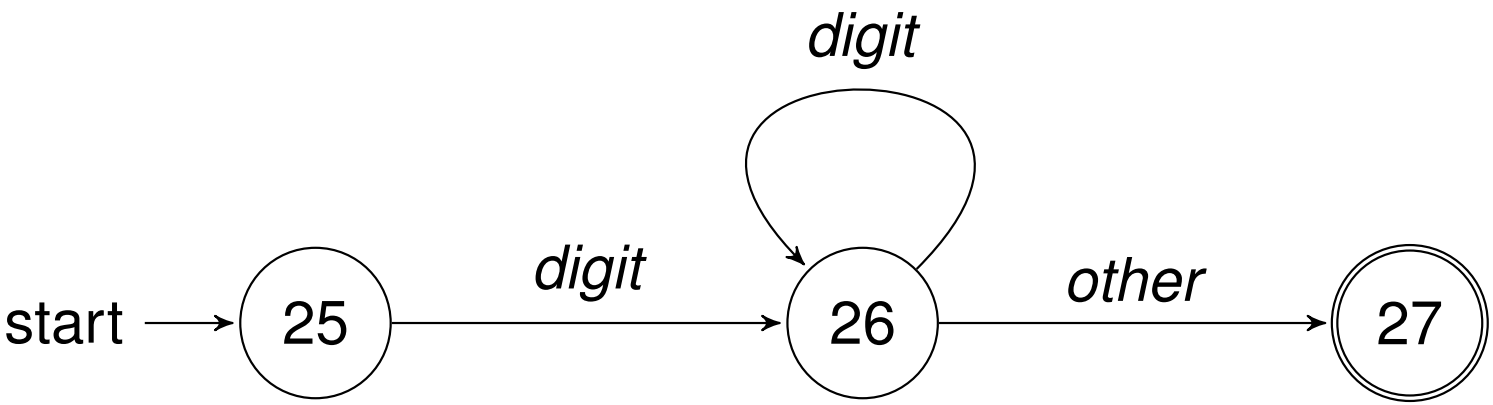
Transition Diagram for Numbers



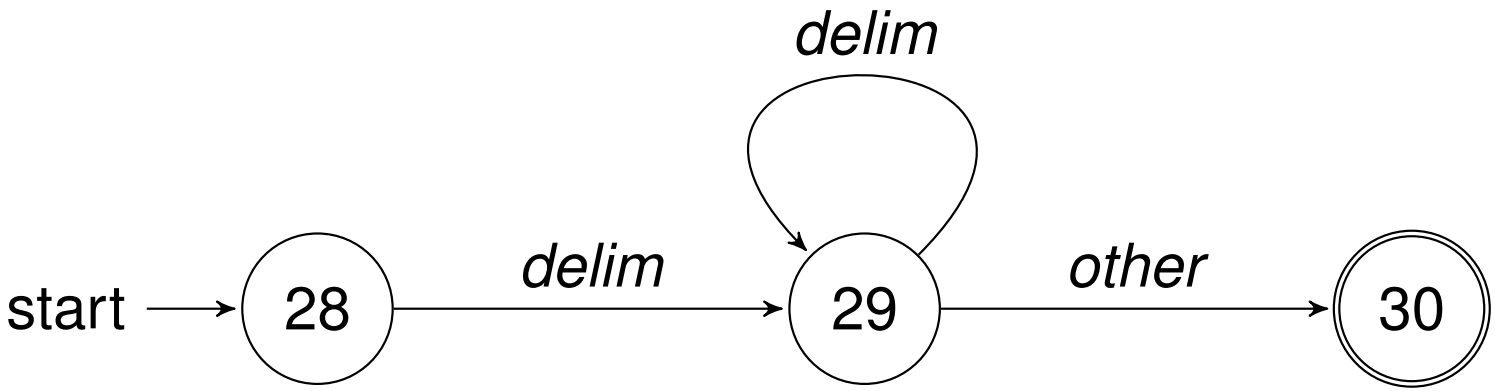
Recognitions of Tokens

Transition Diagram for Numbers





Transition Diagrams for White spaces



Implementing a Transition Diagram

```
int yylex(){
  while(1){
     switch(state){
        case 0: c = nextchar();
                if (c== blank || c== tab || c==newline){
                        state = 0;
                }
                else if (c == '<') state = 1;
                else if (c == '=') state = 5;
                else if (c == '>') state = 6;
                else state = fail();
                break;
                }
          case 1:
          .
          .
          case 9: c = nextchar();
                  if (isletter(c))state = 10;
                  else state = fail();
                  break;
           case 10: c = nextchar();
                    if (isletter(c)) state = 10;
                    else if (isdigit(c)) state = 10;
                    else state = 11;
                    break;
  }
}
```

Assignment

Implement a lexical analyzer for the following types of tokens:

- Arithmetic, Relational, Logical, Bitwise and Assignment Operators of C.
- Reserved words: int, float, char, for, while, if and else
- Identifier.
- Integer Constants.
- Parentheses, Curly braces

Follow the ideas of transition diagram, yytext, yyleng, etc as stated in the study material.