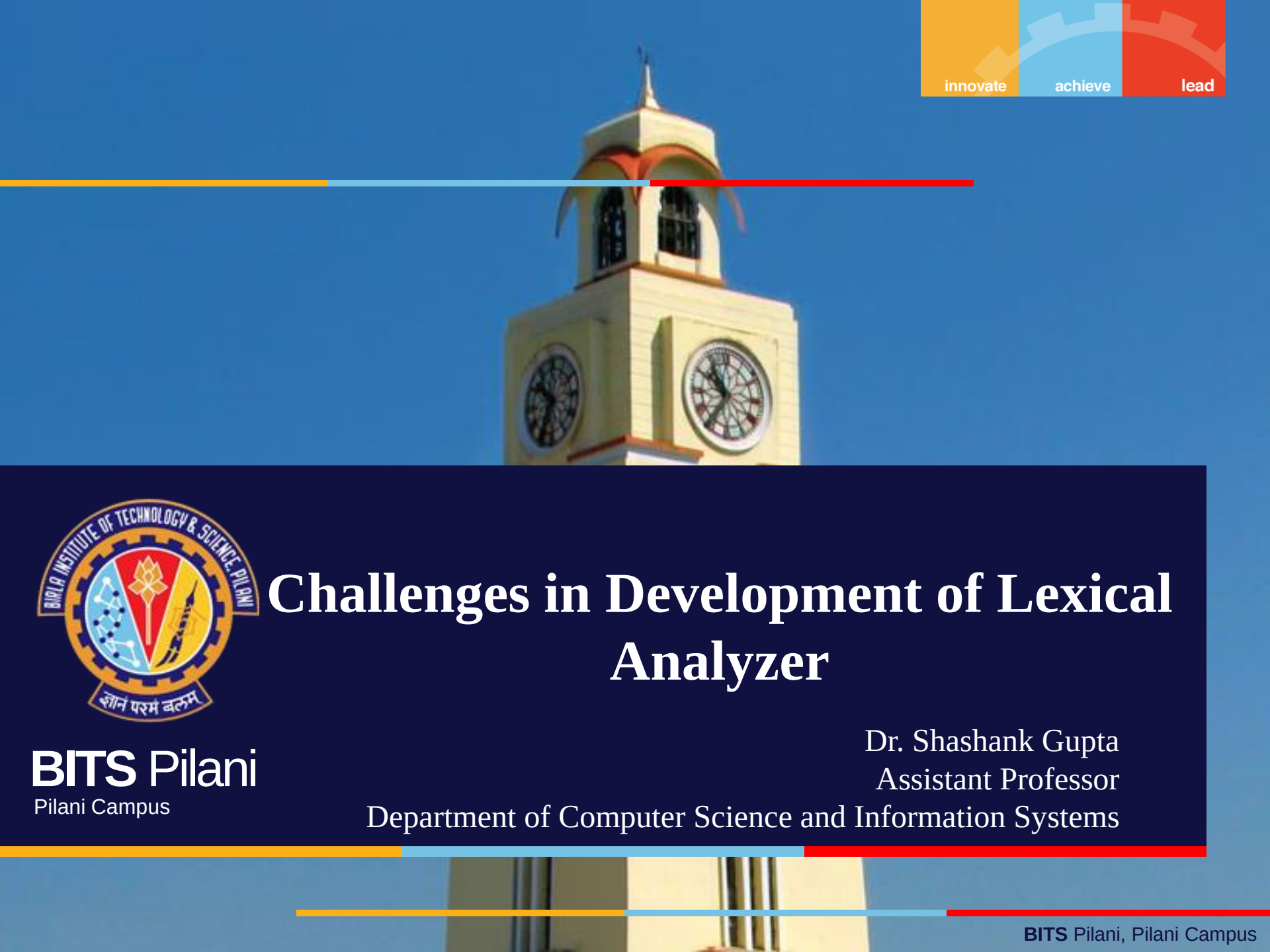


innovate achieve lead

Challenges in Development of Lexical Analyzer

Dr. Shashank Gupta
Assistant Professor
Department of Computer Science and Information Systems

BITS Pilani
Pilani Campus

Development Issues related to Symbol Table

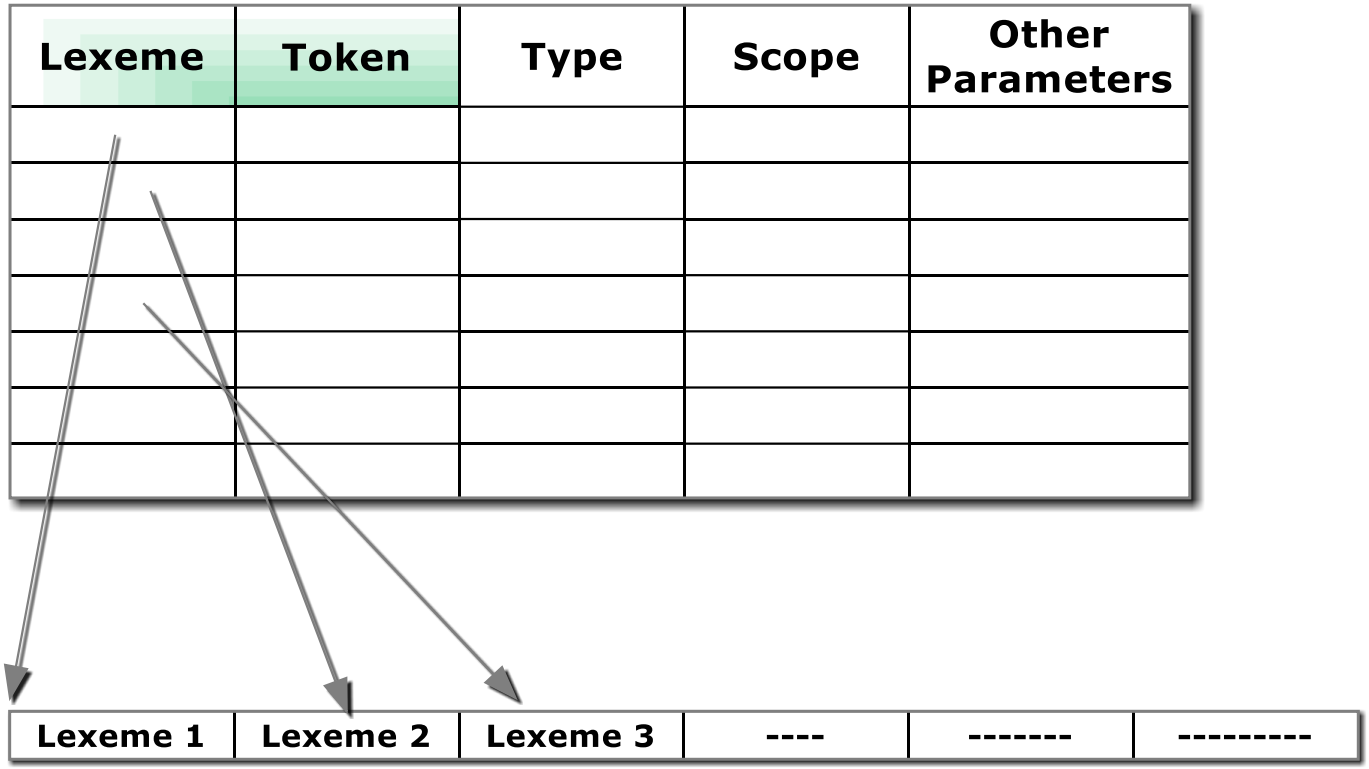
Static quantity of space for saving the lexemes in symbol table is not recommended.

- Since, average size of variable is 7 to 8 characters long.

Instead, save lexemes in a some separate memory. Symbol table has pointers to lexemes.

- This will make the symbol table space efficient.

Development of Symbol Table



Keywords in Lexical Analyzer

Since, the rule for recognizing the Identifier and Keywords are same as L. (L+D)*.

- How to tokenize Identifiers and Keywords separately?

Initialize symbol table with the list of keywords in your language specification as follows:

- `insert( "if" , keyword ) and insert( "MOD" , MODULUS).`

Hence, before inserting anything in the symbol table.

Perform lookup operation and any subsequent lookup returns a nonzero value, therefore, cannot be used as an identifier.

Challenges in Development of Lexical Analysis

Fix Versus Free Lexemes

- `flag = flag * 6;`
- `vs`
- `flag = flag`
- `*6;`

How to Deal with White Spaces ?

- `fl ag = f  la  g  * 6;`

Challenges in Development of Lexical Analysis

Sometimes, languages specifications are not complete.

- How to tokenize the following two inputs?
- `Z20`
- `Z2000`

Such specification issues need to be handled during implementation of lexical analyzer.

Technique for Specifying the Tokens

Tokens are generally best defined by Regular Languages.

Regular languages have a very nice mathematical theory.

- A regular expression `r` denotes a regular language `L(r)`.

Regular expressions are very popular for specifications of tokens.

Regular Definitions

Consider R_i be any regular expression and N_i be any unique name then a regular definition is a series of definition of the following form:

$$N_1 \rightarrow R_1$$
$$N_2 \rightarrow R_2$$
$$N_3 \rightarrow R_3$$
$$-$$
$$-$$
$$N_n \rightarrow R_n$$

where each R_i *is a regular* exp*ression over*

$$\sum \bigcup \{N_1, N_2, N_3, ---N_{i-1}\}$$

Regular Definition for Identifiers

$$Alphabet \rightarrow A|B|C|----|Z|a|b|c|----|z$$

$$Digit \rightarrow 0|1|2|----|9$$

$$Identifier \rightarrow Alphabet\ (Alphabet|Digit)^*$$

Regular Definition for Unsigned Numbers

$$Digit \rightarrow 0|1|2|-----|9$$

$$Digits \rightarrow Digit^{+}$$

$$Fraction \rightarrow '.'Digits|\in$$

$$Exponent \rightarrow (E\,(+|-|\in)\,Digits)|\in$$

$$Number \rightarrow Digits\ Fraction Exponent$$