ZenPro Programming Language

DHA Suffa University

Muneeb Ur Rehman
CS211113

Makia Idrees
CS211278

Introduction:
This document describes the ZenPro language, ZenPro Language is a compiled programming language. It is a basic language for Beginner's programmers as it is easy to learn and understandable. It can be used to learn basic programming concepts.
Extension: zen

Purpose:
The ZenPro language is a small, imperative language with integer and string variables and some if else condition. Its syntax resembles some functional languages. The Motivation behind this ZenPro Language is C language, our main purpose was to create a language that is easy to learn, easy to understandable, compiled and can be used to teach kids at school, colleges or Universities

Lexical Analysis

Keywords are reserved words that convey special message. The reserved words are if, else, do, while, array, break, functions, etc.

An identifier is a sequence of letter followed by digits, underscore. It is always be in camel case.

A numeric constant represent numeric values in various forms like integer and float. Lexeme recognize numeric constant such as 123, 0.456.

Operators performs operation on vales and variable such as arithmetic operators (+, -, <, >, *, /) and assignment operators (=). Lexers distinguish and classify the symbols.

The punctuation symbols are, ( ) { } : ; " " + - <= >= <> / | != += . Lexeme recognize them and classify into different categories.

String literals are sequence of letters and digits surrounded by double quotes (""). White space include spaces, tabs, line breaks. Ignored by compiler but crucial for code readability.

Escape sequences are used to represent special characters within strings or character literals. An escape sequence is a combination of characters that begins with a backslash \ and is followed by one or more characters, indicating a special interpretation. Here are some common escape sequences:

\t New tab
\n new line
\\ Backslash
\ Single quotes
\" Double quotes

Regular Expressions:

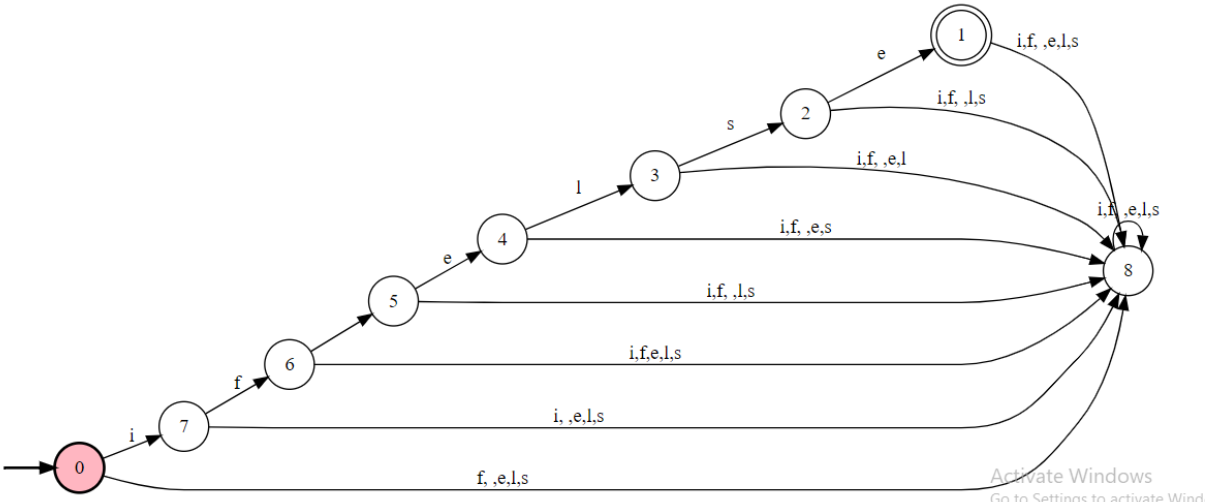
L (operators)
→ operators:- ‘+’, ‘-’, ‘*’, ‘/’, ‘=’, ‘>’, ‘<’, ‘&’, ‘|’, ‘%’, ‘>=’, ‘<=’
L (numbers)*
→ numbers:- ‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’
L(language)*
{“(a-z)(A-Z)”}
Σ = {(a-z), (A-Z)}
L {(“){(letters)*| (numbers)* | (symbol)*} (”)}

L(symbols):
Symbols:- ‘;’ , ‘\l’, ‘, ’, ‘(’, ‘)’, ‘[’, ‘]’, ‘{’, ‘}’, ‘.’

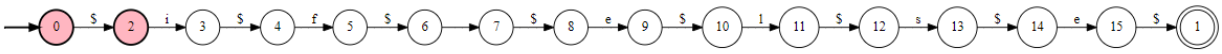
Lexical Units:

LEXEME	*TOKEN*	*REGULAR EXPRESSION*	*PATTERN*
If, else, for, while	YES, No, FOR_CD, WHILE_CD	If, else, for, while	Reserved keywords convey special meaning
Var1, var2, num1….	IDENTIFIER	Var1, var2, num1….	It is a variable
( ) [ ] { } , : ; .	OB, CB..	, or : or ; or (or) or [or] or { or } or .	Punctuation Symbols
<=, >=, ==	COMPARE_OP	< or > or <= or >= or == or !=	Comparison Operator
Int, float, string, char	INT_KW, FLOAT_KW, STRING_KW, CHAR_KW	Int, float, string, char..	Reserved keywords convey special meaning
1, 2, 3….	NUM_CONST	0-9	A constant number
0.1, 0.2...	NUM_CONST	0.1,0.2..	Numbers till 0-9
\n , \t, \\, \ ^ c	ESCAPE_SEQ	\n, \t, \\	Escape sequence is used for spacing in code
“Hello, World!”	STR_LITERAL	"([^"]*)"	String literal surrounded by double quotes(“”)
Print, break, input	SHOW, STOP, WRITE	Print, break, write	Reserved keyword for printing values
= ,+,-,*,/,^,%	OPERATOR	= or + or – or * or / or ^ or %	Arithmetic operator or symbols used on binary digits

DFA



NFA using Thompson Construction



NFA