Documentation

Sunday, January 26, 2014 8:43 PM

1. Description of Lexical Conventions

- Case-sensitivity
    - In this program, we will assume that case is important; therefore, "Bob" would be a different identifier from "bob"
- Removing comments
    - Two types of comments: /* …. */ and //..
    - For /* …. */, remove delimiters plus everything in between
    - For // .., remove delimiters plus everything following the delimiter, until we get to a carriage return
- Handling whitespace
    - whitespace signals end of a token; we also skip all extraneous whitespace
    - Whitespace include: space, tab, carriage return
- Valid characters in the alphabet (19)
    - Letters [a-z, A-Z], numbers [0-9], whitespace (space, tab, carriage return), =, <, >, ;, ",", ., (,), {, }, [,], /, \, *, _
- Definition of Atomic lexical elements
    - id ::= letter (letter|0|nonzero|_)*
    - num ::= nonzero (0|nonzero)* (.(0|nonzero)* nonzero|.0|epsilon) | 0 (.(0|nonzero)* nonzero|.0|epsilon)
    - letter ::= a..z | A..Z
    - nonzero ::= 1..9
- Codes used for tokens (numbers and capital acronyms are normally used)

Token type	Lexeme	Token type	Lexeme
id	handle	and	and
num	1.23	not	not
relop_g	>	or	or
relop_l	<	if	if
relop_ge	>=	then	then
relop_le	<=	else	else
relop_e	==	for	for
assignop	=	class	class
add	+	int	int
sub	-	float	float
mul	*	get	get
div	/	put	put
semi	;	return	return
comma	,	opencur	{
dot	.	closecur	}
openpar	(	opensq	[
closepar	)	closesq	]
comment_long	/*..*/	comment	//..
lege	<>		

- FSM: check FSM diagram PDF
- Error Handling:
    - We will use panic mode recovery for error handling. That is, if an error occurs, the

analyzer will look for the next available token to read. This method is straight-forward and still allows the entire document to be read and tokenized.

- All possible lexical errors that the lexical analyzer might encounter
    - Panic mode recovery: explain reason for choosing this method