Rony Vargas
155005725
CS214 – Systems Programming
Spring 2015

Assignment 1 – Readme

Main function takes one additional argument. This will be a pointer to a string that will be tokenized using the token rules outlined.

The strings are then parsed by the function stringParser that will remove unwanted symbols from the strings. This has to be done because escape character in command line input are treated as two characters. This function will in a sense "fix" arguments passed in from the command line. It works by keeping track of two index's "i" and "k". "i" is an index to each character being checked. "k" index the current position available if the "i" index is valid. A memmove is done if "i" index is valid. A reallocation of memory is done if the string length is now shorter due to removed characters.

A tokenizerT object pointer is then created using the parsed string and then initialized. A tokenizerT consists of a token type, the string that will be tokenized, and two integers, which point to locations in the string. These integers represent the beginning and end of the new token to be returned.

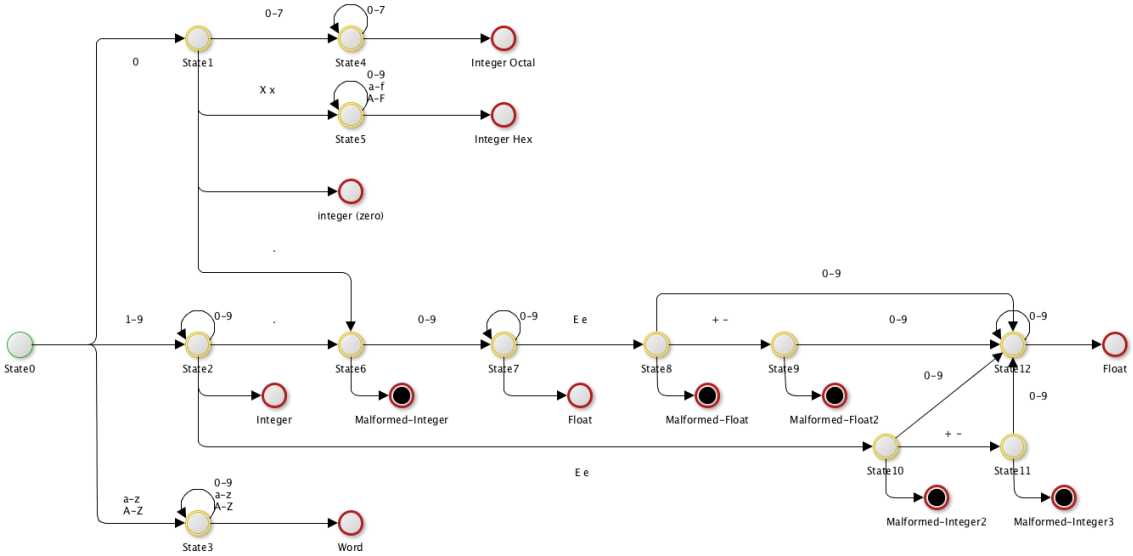
TKGetNextToken will use the function nextState to determine the end and beginning of the tokens by sending the current state and the next character. If a "token" state is returned then it will determine the size of the token, allocate space and return that. The size and proper index of the produced token is stored in the tokenizerT Object as two ints.

The nextState function is based on the state transitions attached below. The nextState takes two arguments, a char and a pointer to a state. Depending on this combination then the function will change the state of the pointer.

On every call to TKGetNextToken it will return a token from the string. The current state of the tokenizer object is passed to printState which will print the token type and then the actual string will be printed by the printToken function.

A few notes on the expected output.
Extra \ will be removed.
\'char' – where char is an escape char will be turned to one charactater
Escape characters will be printed as an error & hex
Malformed tokens are correct and the next token will start with first character that violated token type. Ex 123. Integer 123, full stop .

Rony Vargas
155005725
CS214 – Systems Programming
Spring 2015



Rony Vargas
155005725
CS214 – Systems Programming
Spring 2015

State 1		State 2		State 3	
`	backtick				
~	bitwise complement				
!	negate	!=	not equal to		
@	at sign				
#	octothorpe				
$	dollar sign				
%	modulus	%=	assign remainder		
^	bitwise XOR	^=	assign bitwise XOR		
&	address/bitwise AND	&&	logical AND		
		&=	assign bitwise AND		
*	multiply/dereference	*=	assign product		
?					
(	open parenthesis				
)	close parenthesis				
-	subtract	--	decrement		
		-=	assign difference		
		->	indirect component selector		
_	dash				
=	assign	==	equals to		
+	add	++	increment		
		+=	asign sum		
'	single quote				
:	colon				
;	semi-colin				
"	double quotes				
<	less than	<<	shift left	<<=	assign left shift
		<=	less than or equal to		
>	greater than	>>	shift right	>>=	assign right shift
		>=	greater than or equal to		
,	comma				
.	full stop	..	malformed	...	ellipsis
/	divide	/=	assign quotient		
|	bitwise OR	||	logical or		
		|=	assign bitwise OR		
[	open bracket				
]	close bracket				
{	open curly bracket				
}	close curly bracket				

Rony Vargas
155005725
CS214 – Systems Programming
Spring 2015