Project Part 2

Monday, March 28, 2022 7:07 PM

Cameron S. Williamson

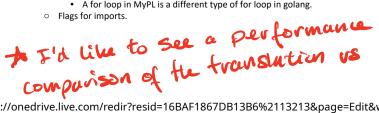
MyPL to Golang

- Package main
- Imports will be required
 - o fmt.Print()
- Weird Behavior:
 - o Getting characters
 - Get(string, int) == string[int]
 - Reading strings
 - Reader := bufio.NewReader(os.Stdin)
 - Fmt.Print("Enter text: ")
 - Text, _ := reader.ReadString('\n')
 - Fmt.Println(text)



Program	(<tdecl> <fdecl>)*</fdecl></tdecl>	(<tdecl> <fdecl>)*</fdecl></tdecl>
Tdecl	TYPE ID LBRACE <vdecls> RBRACE</vdecls>	TYPE ID STRUCT LBRACE <vdecls> RBRACE</vdecls>
Vdecls	(<vdecl_stmt>)*</vdecl_stmt>	(<vdecl_stmt>)*</vdecl_stmt>
Fdecl	FUN (<dtype> VOID) ID LPAREN <params> RPAREN LBRACE <stmts> RBRACE</stmts></params></dtype>	FUN ID LPAREN <params> RPAREN LBRACE <stmts>RBRACE</stmts></params>
Params	<dtype> ID (COMMA <dtype> ID)*ϵ</dtype></dtype>	ID <dtype> (COMMA <dtype> ID)*ϵ</dtype></dtype>
Dtype	INT_TYPE DOUBLE_TYPE BOOL_TYPE CHAR_TYPE STRING_TYPE ID	INT_TYPE DOUBLE_TYPE BOOL_TYPE CHAR_TYPE STRING_TYPE ID
Stmts	(<stmt>)*</stmt>	(<stmt>)*</stmt>
Stmt	<pre><vdecl_stmt> <assign_stmt> <cond_stmt> <while_stmt> <for_stmt> <call_expr> <ret_stmt> <delete_stmt></delete_stmt></ret_stmt></call_expr></for_stmt></while_stmt></cond_stmt></assign_stmt></vdecl_stmt></pre>	<pre></pre> <pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre><pre><pre><pre><pre><pre><pre><pre><pre><pre><pre><pre><pre><pre><pre><</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
Vdecl_stm t	VAR (<dtype> ε) ID ASSIGN <expr></expr></dtype>	(VAR ID dtype ASSIGN <expr>) (ID COLON ASSIGN <expr>)</expr></expr>
Assign_st mt	<lvalue> ASSIGN <expr></expr></lvalue>	<lu>alue> ASSIGN <expr></expr></lu>
Lvalue	ID (DOT ID)*	ID (DOT ID)*
Cond_stmt	IF <expr> LBRACE <stmts> RBRACE <condt></condt></stmts></expr>	IF <expr> LBRACE <stmts> RBRACE <condt></condt></stmts></expr>
Condt	ELIF <expr> LBRACE <stmts> RBRACE <condt> ELSE LBRACE <stmts> RBRACE ϵ</stmts></condt></stmts></expr>	ELIF <expr> LBRACE <stmts> RBRACE <condt> ELSE LBRACE <stmts> RBRACE ϵ</stmts></condt></stmts></expr>
While_stm t	WHILE <expr> LBRACE <stmts> RBRACE</stmts></expr>	FOR <expr> LBRACE <stmts> RBRACE</stmts></expr>
For_stmt	FOR ID FROM <expr> (UPTO DOWNTO) <expr> LBRACE <stmts> RBRACE</stmts></expr></expr>	FOR <vdecl_stmt> SEMICOLON <expr> <expr> R</expr></expr></vdecl_stmt>
Call_expr	ID LPAREN <args> RPAREN</args>	ID LPAREN <args> RPAREN</args>
Args	<expr> (COMMA <expr>)* ϵ</expr></expr>	<expr> (COMMA <expr>)* ϵ</expr></expr>
Ret_stmt	RETURN ($\langle expr \rangle \mid \epsilon$)	RETURN (<expr> ϵ</expr>
Delete_st mt	DELETE ID	DNE
Expr	(<rvalue> NOT <expr> LPAREN <expr> RPAREN) (<operator> <expr> ϵ)</expr></operator></expr></expr></rvalue>	
Operator	PLUS MINUS DIVIDE MULTIPLY MODULO AND OR EQUAL LESS_THAN GREATER_THAN LESS_THAN_EQUAL GREATER_THAN_EQUAL NOT_EQUAL	
Rvalue	<pval> NIL NEW ID <idrval> <call_expr> NEG <expr></expr></call_expr></idrval></pval>	
Pval	INT_VAL DOUBLE_VAL BOOL_VAL CHAR_VAL STRING_VAL	
Idrval	ID (DOT ID)*	

- · Things I need to watch out for
 - Converting the loops
 - A while loop in MyPL is a for loop in golang.
 - A for loop in MyPL is a different type of for loop in golang.



since the translation will incor Some over head