Compiler Construction: Introduction

Project D

Eugene Zouev

Fall Semester 2018

Innopolis University

Project D: Dynamic language

- Object types are not specified and can change while program execution
- The language is interpreted
- Major notion: variable & literal (constant)
- Program structure: a sequence of declarations and statements
- Types:
  built-in: integer, real, boolean, string
  user-defined: array, tuple, function
- Implicit type conversions
- Statements: assignment, if/while, return, input/output

Project D: the Language Syntax

```
Program     : { Declaration }

Declaration : var Identifier [ := Expression ] ;
```

```
Expression  : Relation [ ( and | or | xor ) Relation ]

Relation    : Factor [ ( < | <= | > | >= | = | /= ) Factor ]

Factor      : Term { ( + | - ) Term }

Term        : Unary { ( * | / ) Unary }

Unary       : [ + | - | not ] Primary [ is TypeIndicator ]
            | Literal
            | ( Expression )

Primary     : Identifier { Tail }
            | readInt | readReal | readString

Tail    : . IntegerLiteral   // access to unnamed tuple element
        | . Identifier       // access to named tuple element
        | [ Expression ]     // access to array element
        | ( Expression { , Expression } )  // function call
```

Project D: the Language Syntax

```
Statement   : { Assignment | Print | Return | If | Loop }

Assignment  : Primary := Expression

Print       : print Expression { , Expression }

Return      : return [ Expression ]

If          : if Expression then Body [ else Body ] end

Loop        : while Expression LoopBody
            | for Identifier in TypeIndicator LoopBody

LoopBody    : loop Body end

TypeIndicator : int | real | bool | string
              | empty              // no type
              | [ ]                // vector type
              | { }                // tuple type
              | func               // functional type
              | Expression .. Expression
```

Project D: the Language Syntax

```
Literal : IntegerLiteral   // 1, 12345, 777
        | RealLiteral      // 1.23
        | BooleanLiteral   // true or false
        | StringLiteral    // "string", 'string‘
        | ArrayLiteral
        | TupleLiteral

ArrayLiteral : [ [ Expression { , Expression } ] ]

TupleLiteral : { [ [ Identifier := ] Expression
                         { , [ Identifier := ] Expression } ] }

FunctionLiteral : func [ Parameters ] FunBody

Parameters      : ( Identifier { , Identifier } )

FunBody         : is Body end
                | => Expression

Body            : { Declaration | Statement | Expression }
```

Project D: Operators

Sign	Operand1	Operand2	Result	Semantics
+	Integer	Integer	Integer	Algebraic addition
	Integer	Real	Real	
	Real	Integer	Real	
	Real	Real	Real	
	Vector	Integer	Vector	Adding element to vector
	Vector	Real	Vector	
	Vector	Vector	Vector	Joining vectors
	List	Integer	List	Appending element to list
	List	Real	List	
	List	List	List	Joining lists
+=	Integer	Integer	Integer	
	Tuple	Tuple	Tuple	Memberwise addition
-	Integer	Integer	Integer	Algebraic subtraction
	Integer	Real	Real	
	Real	Integer	Real	
	Real	Real	Real	

Project D: Implementation Model