

CP471 Assignment 2

Aaron Exley

February 14, 2020

LL(1) Language

```
<program> ::= <fdecls> <declarations> <statement_seq>.
<fdecls> ::= <fdec>; <fdecls> | ε
<fdec> ::= def <type> <fname> ( <params> ) <declarations> <statement_seq> fed
<params> ::= <type> <var><params_rest> |
<params_rest> ::= , <params> | ε
<fname> ::= <id>

<declarations> ::= <decl>; <deklarations> | ε
<decl> ::= <type> <varlist>
<type> ::= int | double
<varlist> ::= <var><varlist_rest>
<varlist_rest> ::= , <varlist> | ε

<statement_seq> ::= <statement><statement_seq_rest>
<statement_seq_rest> ::= ; <statement_seq> | ε

<statement> ::= <var> = <expr> |
               if <bexpr> then <statement_seq> <if_rest> |
               while <bexpr> do <statement_seq> od |
               print <expr> |
               return <expr> |
<if_rest> ::= fi | else <statement_seq> fi |

<expr> ::= <term><expr_rest>
<expr_rest> ::= + <term><expr_rest> | - <term><expr_rest> | ε
<term> ::= <factor><term_rest>
<term_rest> ::= * <factor><term_rest> | / <factor><term_rest> | % <factor><term_rest> | ε
<factor> ::= <var> | <number> | (<expr>) | <fname>(<exprseq>)
<exprseq> ::= <expr><exprseq_rest>
<exprseq_rest> ::= , <exprseq> | ε

<bexpr> ::= <bterm><bexpr_rest>
<bexpr_rest> ::= or <bterm><bexpr_rest> | ε
<bterm> ::= <bfactor><bterm_rest>
<bterm_rest> ::= and <bfactor><bterm_rest> | ε
<bfactor> ::= (<bfactor_rest>) | not <bfactor>
<bfactor_rest> ::= <bexpr> | <expr> <comp> <expr>

<comp> ::= < | > | == | <= | >= | <>

<var> ::= <id> | <id>[<expr>]

<letter> ::= a | b | c | ... | z
<digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0
<id> ::= <letter> | <id><letter> | <id><digit>

<number> ::= <integer> | <double>
```

First

$\text{First}(\langle \text{program} \rangle) = \{\text{def}, \}$

$\text{First}(\langle \text{fdecls} \rangle) = \{\text{def}, \epsilon\}$

$\text{First}(\langle \text{fdec} \rangle) = \{\text{def}\}$

$\text{First}(\langle \text{params} \rangle) = \{\text{int}, \text{double}, \}$

$\text{First}(\langle \text{params_rest} \rangle) = \{, , \epsilon\}$

$\text{First}(\langle \text{fname} \rangle) = \{\text{id}\}$

$\text{First}(\langle \text{declarations} \rangle) = \text{First}(\langle \text{decl} \rangle) = \text{First}(\langle \text{type} \rangle) = \{\text{int}, \text{double}\}$

$\text{First}(\langle \text{varlist} \rangle) = \{\text{id}\}$

$\text{First}(\langle \text{varlist_rest} \rangle) = \{, , \epsilon\}$

$\text{First}(\langle \text{statement_seq} \rangle) = \text{First}(\langle \text{statement} \rangle) = \{\text{id}, \text{if}, \text{while}, \text{print}, \text{return}\}$

$\text{First}(\langle \text{statement_seq_rest} \rangle) = \{; , \epsilon\}$

$\text{First}(\langle \text{if_rest} \rangle) = \{\text{fi}, \text{else}\}$

$\text{First}(\langle \text{expr} \rangle) = \text{First}(\langle \text{term} \rangle) = \text{First}(\langle \text{factor} \rangle) = \text{First}(\langle \text{exprseq} \rangle) = \{\text{id}, \text{integer}, \text{double}, (\}$

$\text{First}(\langle \text{expr_rest} \rangle) = \{+, -, \epsilon\}$

$\text{First}(\langle \text{term_rest} \rangle) = \{*, /, \%, \epsilon\}$

$\text{First}(\langle \text{exprseq_rest} \rangle) = \{, , \epsilon\}$

$\text{First}(\langle \text{bexpr} \rangle) = \text{First}(\langle \text{bterm} \rangle) = \text{First}(\langle \text{bfactor} \rangle) = \{(\text{, not}\}$

$\text{First}(\langle \text{bexpr_rest} \rangle) = \{(\text{or}, \epsilon\}$

$\text{First}(\langle \text{bterm_rest} \rangle) = \{(\text{and}, \epsilon\}$

$\text{First}(\langle \text{bfactor_rest} \rangle) = \{\text{id}, \text{integer}, \text{double}, (\text{, not}\}$

$\text{First}(\langle \text{comp} \rangle) = \{<, >, ==, <=, >=, <>\}$

$\text{First}(\langle \text{var} \rangle) = \{\text{id}\}$

Follow