CP414 Assignment 1

Aaron Exley

January 15, 2020

```
<program> ::= <fdecls> <declarations> <statement seq>.
<fdecls> ::= <fdec>; | <fdecls> <fdec>; |
<fdec> ::= def <type> <fname> ( <params> ) <declarations> <statement seq> fed
<params> ::= <type> <var> <params rest> |
<params_rest> ::= , <params> | \epsilon
<fname> ::= <id>
<declarations> ::= <decl>; | <declarations> <decl>; |
<decl> ::= <type> <varlist>
\langle \text{type} \rangle ::= \text{int} \mid \text{double}
<varlist> ::= <var> <varlist\_rest>
\langle \text{varlist} \mid \text{rest} \rangle ::=, \langle \text{varlist} \rangle \mid \epsilon
<statement_seq> ::= <statement><statement_seq_rest>
\langle \text{statement seq rest} \rangle ::= ; \langle \text{statement seq} \rangle \mid \epsilon
\langle \text{statement} \rangle ::= \langle \text{var} \rangle = \langle \text{expr} \rangle
                        if <bexpr> then <statement seq> <if rest> |
                        while <br/> do <statement_seq> od |
                        print <expr> |
                        return <expr> |
\langle \text{if rest} \rangle ::= \text{fi} \mid \text{else} \langle \text{statement seq} \rangle \text{fi} \mid
\langle expr \rangle ::= \langle term \rangle \langle expr\_rest \rangle
\langle \expr_rest \rangle ::= + \langle term \rangle \langle \expr_rest \rangle | - \langle term \rangle \langle \expr_rest \rangle | \epsilon
<term> ::= <factor> <term rest>
<term_rest> ::= * <factor> <term_rest> | / <factor> <term_rest> | % <factor> <term_rest> | \epsilon
<factor> ::= <var> | <number> | (<expr>) | <fname>(<exprseq>)
\langle \text{exprseq} \rangle ::= \langle \text{expr} \rangle \langle \text{exprseq} \text{ rest} \rangle
<exprseq_rest> ::= , <exprseq> | \epsilon
<bexpr> ::= <bexpr> <bexpr> =
\langle \text{bexpr rest} \rangle ::= \text{or } \langle \text{bexpr rest} \rangle \mid \epsilon
<bterm> ::= <bfactor> <bterm rest>
\langle \text{bterm rest} \rangle ::= \text{and } \langle \text{bfactor} \rangle \langle \text{bterm rest} \rangle \mid \epsilon
<bfactor> ::= (<bfactor_rest>) | not <bfactor>
<bfactor rest> ::= <bexpr> | <expr> <comp> <expr> |
<comp> ::= < | > | == | <= | >= | <>
\langle var \rangle ::= \langle id \rangle | \langle id \rangle [\langle expr \rangle]
<letter> ::= a | b | c | ... | z
<digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0
\langle id \rangle ::= \langle letter \rangle \mid \langle id \rangle \langle letter \rangle \mid \langle id \rangle \langle digit \rangle
<number> ::= <integer> | <double>
```

```
First and Follow First(\langle program \rangle) = {def, }
First(<fdecls>) = \{def, \epsilon\}
First(\langle fdec \rangle) = \{def\}
First(<params>) = \{int, double,\}
First(< params_rest) = \{,, \epsilon \}
First(\langle fname \rangle) = \{id\}
First(<decl>) = First(<decl>) = First(<type>) = {int, double}
First(\langle varlist \rangle) = \{id\}
First(\langle varlist\_rest \rangle) = \{,, \epsilon\}
First(\langle statement \_seq \rangle) = First(\langle statement \rangle) = \{id, if, while, print, return\}
First(\langle statement\_seq\_rest \rangle) = \{;, \epsilon\}
First(\langle if\_rest \rangle) = \{fi, else\}
First(\langle expr \rangle) = First(\langle factor \rangle) = First(\langle exprseq \rangle) = \{id, integer, double, (\}\}
First(\langle expr\_rest \rangle) = \{+, -, \epsilon\}
\begin{aligned} & \text{First}(<\text{term\_rest}>) = \{*, /, \%, \epsilon\} \\ & \text{First}(<\text{exprseq\_rest}>) = \{, , \epsilon\} \end{aligned}
First(\langle bexpr \rangle) = First(\langle bfactor \rangle) = \{(, not\}\}
First(\langle bexpr\_rest \rangle) = \{(or, \epsilon\}\}
First(\langle bterm\_rest \rangle) = \{(and, \epsilon\}\}
First(\langle bfactor\_rest \rangle) = \{id, integer, double, (, not\}\}
First(<comp>) = \{<, >, ==, <=, >=, <>\}
First(\langle var \rangle) = \{id\}
```