3.3. Arith. exp.

3.3.1. Syntax

✓ <add operator> ::= + | − ✓ <multop> ::= × | / | ÷

procedure
<function value>

✓ primary
<prime exp> ::= <unsigned number> | <variable> | (<arithmetic expression>)

✓ <factor> ::= <prime exp> | <factor> ↑ <prime exp>

✓ <term> ::= <factor> | <term> <multop> <factor>

(Rot) ~~<simple term>~~ ::=
<term sum> ::= <term> | <add op> <term> | <term sum> <add op> <term>

<~~simple~~ arithm. exp.> ::= IF <boolean ~~exp~~ expression> then <term sum>

major
<major arithmetic expression> ::= <~~major~~ simple arithm. exp> | <major arithmetic expression> else <simple arithm. exp>

<arithmetic expression> ::= <term sum> | <major arithm. exp> | <major arithm. exp> else <term sum>

<relation> ::= <prime exp> <relational operator> <prime exp>

<b prime> ::= <relation> | ¬ <relation> | (<boolean expression>) | true | false | <variable> | <function value>

<b factor> ::= <b prime> | <b factor> ∧ <b prime>

<b term> ::= <b factor> | <b term> ∨ <b factor>

<imp boolean> ::= <b term> | <imp boolean> ⊃ <b term>

<minor boolean> ::= <imp boolean> | <minor boolean> ≡ <imp boolean>

<simple boolean> ::= ~~<minor boolean>~~ IF <boolean expression> then <minor boolean>

<boolean> ::= <simple boolean> | <boolean> else <simple boolean>

<boolean expression> ::= <boolean> | <boolean> else <~~simple~~ minor boolean> | <minor boolean>

<d prime> ::= <label> | <switch value>

<major designational expression> ::= <d prime> | IF <boolean expression> then <major designational expression>

<designational expression> ::= <major designational expression> |
<designational expression> else <major designational expression>

<arithmetic assignment statement> ::= <Variable> := <arithmetic expression> |

<Variable> := <arithmetic assignment statement>

<boolean assignment statement> ::= <Variable> := <boolean expression> |

<Variable> := <boolean assignment statement>

<assignment statement> ::= <arithmetic assignment statement> |

<boolean assignment statement>

<unlabelled basic statement> ::= <assignment statement> | <goto statement> | <procedure statement> |

~~<do statement>~~ | <dummy statement>

<basic statement> ::= <unlabelled basic statement> | <Label> : <basic statement>

<compound end> ::= <statement> end | <statement> ; <compound end>

<compound tail> ::= <compound end> | <compound end> <$S_;$> | <compound end> <S_{end}>

<block head> ::= begin <declaration> | <block head> ; <declaration>

<unlabelled compound> ::= begin <compound tail>

<~~simple~~ unlab block> ::= <block head> <compound tail>

<compound> ::= <unlabelled compound> | <label> : <unlabelled compound>

~~<unlabelled block> ::= <for statement> | block <simple block>~~

<block> ::= <unlabelled block> | <label> : <unlabelled block>

<unconditional statement> ::= <basic statement> | <compound> | <block> | <for statement>

<simple conditional> ::= ~~<unconditional statement>~~ if <boolean expression> then <unconditional statement>

<conditional> ::= <simple conditional> | <conditional> else <simple conditional statement>

<conditional statement> ::= <conditional> | <conditional> else <unconditional statement> |

<unconditional statement>

<statement> ::= <unconditional statement> | <conditional statement>