Πανεπιστήμιο Ιωαννίνων
Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ και Πληροφορικής
"Μεταφραστές"
Διδάσκων: Γ. Μανής
Φεβρουάριος 2020

## *Γραμματική της minimal++*

```
<program>                  ::=      program id { <block> }

<block>                    ::=      <declarations> <subprograms> <statements>

<declarations>             ::=      (declare <varlist>;)*

<varlist>                  ::=      ε | id  ( , id )*

<subprograms>              ::=      (<subprogram>)*

<subprogram>               ::=      function id <funcbody>  | procedure id  <funcbody>

<funcbody>                 ::=      <formalpars> { <block> }

<formalpars>               ::=      ( <formalparlist> )

<formalparlist>            ::=      <formalparitem>  ( , <formalparitem> )* | ε

<formalparitem>            ::=      in id | inout id

<statements>               ::=      <statement> | { <statement>  ( ; <statement> )* }

<statement>                ::=      <assignment-stat> |

                                    <if-stat> |

                                    <while-stat> |

                                    <doublewhile-stat> |

                                    <loop-stat> |

                                    <exit-stat> |

                                    <forcase-stat> |

                                    <incase-stat> |

                                    <call-stat> |

                                    <return-stat> |

                                    <input-stat> |

                                    <print-stat>

<assignment-stat>          ::=      id := <expression>

<if-stat>                  ::=      if (<condition>) then <statements> <elsepart>

<elsepart>                 ::=      ε | else <statements>

<while-stat>               ::=      while (<condition>) <statements>

<doublewhile-stat>         ::=      doublewhile (<condition>) <statements>

                                       else <statements>

<loop-stat>                ::=      loop <statements>

<exit-stat>                ::=      exit
```

```
<forcase-stat>           ::=     forcase
                                    ( when (<condition>) : <statements> )*
                                    default: <statements>
<incase-stat>            ::=     incase
                                    ( when (<condition>) : <statements> )*
<return-stat>            ::=     return <expression>
<call-stat>              ::=     call id(<actualpars>)
<print-stat>             ::=     print (<expression>)
<input-stat>             ::=     input (id)
<actualpars>             ::=     ( <actualparlist> )
<actualparlist>          ::=     <actualparitem> ( , <actualparitem> )* | ε
<actualparitem>          ::=     in <expression> | inout id
<condition>              ::=     <boolterm> (or <boolterm>)*
<boolterm>               ::=     <boolfactor> (and <boolfactor>)*
<boolfactor>             ::=     not [<condition>] | [<condition>] |
                                          <expression> <relational-oper> <expression>
<expression>             ::=     <optional-sign> <term> ( <add-oper> <term>)*
<term>                   ::=     <factor>  (<mul-oper> <factor>)*
<factor>                 ::=     constant | (<expression>) | id <idtail>
<idtail>                 ::=     ε | <actualpars>
<relational-oper>        ::=     = | <= | >=  | > | < | <>
<add-oper>               ::=     + | -
<mul-oper>               ::=     * | /
<optional-sign>          ::=     ε | <add-oper>
```