BNF for Core

<prog> ::= *program* <decl seq>   (1)
            begin <stmt seq> *end*
<decl seq> ::= <decl> | <decl> <decl seq>   (2)
<stmt seq> ::= <stmt> | <stmt> <stmt seq>   (3)
<decl> ::= *int* <id list>;   (4)
<id list> ::= <id> | <id>, <id list>   (5)
<stmt> ::= <assign>|<if>|<loop>|<in>|<out> (6)
<assign> ::=<id> = <exp>;   (7)
<if> ::= *if* <cond> *then* <stmt seq> *end;*   (8)
    |*if* <cond> *then* <stmt seq> *else* <stmt seq> *end;*
<loop>   ::= *while* <cond> *loop* <stmt seq> *end;* (9)
<in> ::= *read* <id list>;   (10)
<out>::= *write* <id list>;   (11)

BNF for Core (contd.)

<cond> ::=<comp>|!<cond>   (12)
   | [<cond> && <cond>] | [<cond> *or* <cond>]
<comp> ::= (<op> <comp op> <op>)   (13)
<exp> ::= <trm>|<trm>+<exp>|<trm>-<exp>(14)
<trm> ::= <op> | <op> * <trm>   (15)
<op>   ::= <no> | <id> | (<exp>)   (16)
<comp op> ::= != | == | < | > | <= | >=   (17)
<id>   ::= <let> | <let><id> | <let><no> (18)
<let>   ::= A | B | C | ... | X | Y | Z   (19)
<no>   ::= <digit> | <digit><no>   (20)
<digit> ::= 0 | 1 | 2 | 3 | ... | 9   (21)

- Notes:
  Productions (18)-(21) have no *semantic* significance;
  (19) and (21) are superseded by (19') and (21') on next page:

BNF for Core (contd.)

<let>   ::= A | B | C | D | E | F | G | H | I | J | K | L | M
 | N | O | P | Q | R | S | T | U | V | W | X | Y | Z   (19')

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9   (21')