Grammaire

- PROGRAM ::= program ID ; BLOCK .
- BLOCK ::= CONSTS VARS INSTS
- CONSTS ::= const ID = NUM ; { ID = NUM ; } | ε
- VARS ::= var ID { , ID } ; | ε
- INSTS ::= begin INST { ; INST } end
- INST ::= AFFEC | SI | TANTQUE | ECRIRE | LIRE | ε
- AFFEC ::= ID := EXPR
- SI ::= if COND then INST [else INST]
- TANTQUE ::= while COND do INST
- ECRIRE ::= write (EXPR { , EXPR })
- LIRE ::= read (ID { , ID })
- COND ::= EXPR RELOP EXPR
- RELOP ::= = | <> | < | > | <= | >=
- EXPR ::= TERM { ADDOP TERM }
- ADDOP ::= + | -
- TERM ::= FACT { MULOP FACT }
- MULOP ::= * | /
- FACT ::= ID | NUM | (EXPR)