1 GNU MathProg EBNF

⟨*model*⟩ ::= {⟨*stmt*⟩} [‘`end;`’]

⟨*stmt*⟩ ::= ⟨*set-stmt*⟩ | ⟨*param-stmt*⟩ | ⟨*var-stmt*⟩ | ⟨*ctr-stmt*⟩ | ⟨*obj-stmt*⟩

⟨*set-stmt*⟩ ::= ‘`set`’ ⟨*ident*⟩ [⟨*string-lit*⟩] [⟨*ind-expr*⟩] {[‘`,`’] ⟨*set-att*⟩} ‘`;`’

⟨*param-stmt*⟩ ::= ‘`param`’ ⟨*ident*⟩ [⟨*string-lit*⟩] [⟨*ind-expr*⟩] {[‘`,`’] ⟨*param-att*⟩} ‘`;`’

⟨*var-stmt*⟩ ::= ‘`var`’ ⟨*ident*⟩ [⟨*string-lit*⟩] [⟨*ind-expr*⟩] {[‘`,`’] ⟨*var-att*⟩} ‘`;`’

⟨*ctr-stmt*⟩ ::= [‘`subject`’ ‘`to`’ | ‘`subj`’ ‘`to`’ | ‘`s.t.`’] ⟨*ident*⟩ [⟨*string-lit*⟩] [⟨*ind-expr*⟩] ‘`:`’ ⟨*ctr-def*⟩ ‘`;`’

⟨*ctr-def*⟩ ::= ⟨*num-expr*⟩ [‘`,`’] ‘>=’ ⟨*lin-expr*⟩ [‘`,`’] ‘>=’ ⟨*num-expr*⟩
| ⟨*num-expr*⟩ [‘`,`’] ‘<=’ ⟨*lin-expr*⟩ [‘`,`’] ‘<=’ ⟨*num-expr*⟩
| ⟨*lin-expr*⟩ [‘`,`’] (‘>=’ | ‘<=’ | ‘=’) ⟨*lin-expr*⟩

⟨*obj-stmt*⟩ ::= (‘`maximize`’ | ‘`minimize`’) ⟨*ident*⟩ [⟨*string-lit*⟩] [⟨*ind-expr*⟩] ‘`:`’ ⟨*lin-expr*⟩ ‘`;`’

⟨*set-att*⟩ ::= ‘`dimen`’ ⟨*dimen*⟩ | (‘`within`’ | ‘`:=`’ | ‘`default`’) ⟨*set-expr*⟩

⟨*dimen*⟩ ::= ⟨*digit*⟩ | ‘`1`’ ⟨*digit*⟩ | ‘`20`’;

⟨*param-att*⟩ ::= ‘`integer`’ | ‘`binary`’ | ‘`symbolic`’
| (‘<’ | ‘<=’ | ‘>’ | ‘>=’ | ‘==’ | ‘=’ | ‘<>’ | ‘`!=`’ | ‘`:=`’ | ‘`default`’) ⟨*simple-expr*⟩
| ‘`in`’ ⟨*set-expr*⟩

⟨*var-att*⟩ ::= ‘`integer`’ | ‘`binary`’ | (‘<=’ | ‘>=’ | ‘==’ | ‘=’) ⟨*num-expr*⟩

⟨*expr*⟩ ::= ⟨*num-expr*⟩ | ⟨*sym-expr*⟩ | ⟨*set-expr*⟩ | ⟨*logic-expr*⟩ | ⟨*lin-expr*⟩

⟨*unsubscr-ref*⟩ ::= ⟨*ident*⟩

⟨*subscr-ref*⟩ ::= ⟨*ident*⟩ ⟨*subscript*⟩

⟨*simple-expr*⟩ ::= ⟨*num-expr*⟩ | ⟨*sym-expr*⟩

⟨*subscript*⟩ ::= ‘`[`’ ⟨*simple-expr*⟩ {‘`,`’ ⟨*simple-expr*⟩} ‘`]`’

⟨*tuple*⟩ ::= ‘`(`’ ⟨*simple-expr*⟩ {‘`,`’ ⟨*simple-expr*⟩} ‘`)`’

⟨*num-expr*⟩ ::= ⟨*cond-num-expr*⟩ | ⟨*num-expr-6*⟩

⟨*cond-num-expr*⟩ ::= ‘`if`’ ⟨*logic-expr*⟩ ‘`then`’ ⟨*num-expr-6*⟩ [‘`else`’ ⟨*num-expr-6*⟩]

⟨*num-expr-6*⟩ ::= ⟨*num-expr-5*⟩ {(‘+’ | ‘−’ | ‘`less`’) ⟨*num-expr-5*⟩}

⟨*num-expr-5*⟩ ::= ⟨*iter-num-expr*⟩ | ⟨*num-expr-4*⟩

⟨*iter-num-expr*⟩ ::= (‘sum’ | ‘prod’ | ‘min’ | ‘max’) ⟨*ind-expr*⟩ ⟨*num-expr-4*⟩

⟨*num-expr-4*⟩ ::= ⟨*num-expr-3*⟩ {(‘*’ | ‘/’ | ‘div’ | ‘mod’) ⟨*num-expr-3*⟩}

⟨*num-expr-3*⟩ ::= ⟨*num-expr-2*⟩ | (‘+’ | ‘−’) ⟨*num-expr-2*⟩

⟨*num-expr-2*⟩ ::= ⟨*num-expr-1*⟩ {(‘**’ | ‘^’) ⟨*num-expr-1*⟩}

⟨*num-expr-1*⟩ ::= ⟨*num-ref*⟩ | ⟨*num-func-ref*⟩ | ⟨*num-lit*⟩ | ‘(’ ⟨*num-expr*⟩ ‘)’ | ⟨*num-expr*⟩

⟨*num-ref*⟩ ::= ⟨*subscr-ref*⟩ | ⟨*unsubscr-ref*⟩

⟨*num-func-ref*⟩ ::= ‘abs(’ ⟨*num-expr*⟩ ‘)’
| ‘atan(’ ⟨*num-expr*⟩ ‘)’
| ‘atan(’ ⟨*num-expr*⟩ ‘,’ ⟨*num-expr*⟩ ‘)’
| ‘card(’ ⟨*set-expr*⟩ ‘)’
| ‘ceil(’ ⟨*num-expr*⟩ ‘)’
| ‘cos(’ ⟨*num-expr*⟩ ‘)’
| ‘exp(’ ⟨*num-expr*⟩ ‘)’
| ‘floor(’ ⟨*num-expr*⟩ ‘)’
| ‘gmtime()’
| ‘length(’ ⟨*sym-expr*⟩ ‘)’
| ‘log(’ ⟨*num-expr*⟩ ‘)’
| ‘log10(’ ⟨*num-expr*⟩ ‘)’
| ‘max(’ ⟨*num-expr*⟩ {‘,’ ⟨*num-expr*⟩} ‘)’
| ‘min(’ ⟨*num-expr*⟩ {‘,’ ⟨*num-expr*⟩} ‘)’
| ‘round(’ ⟨*num-expr*⟩ [‘,’ ⟨*num-expr*⟩] ‘)’
| ‘sin(’ ⟨*num-expr*⟩ ‘)’
| ‘sqrt(’ ⟨*num-expr*⟩ ‘)’
| ‘str2time(’ ⟨*sym-expr*⟩ ‘,’ ⟨*sym-expr*⟩ ‘)’
| ‘trunc(’ ⟨*num-expr*⟩ [‘,’ ⟨*num-expr*⟩] ‘)’
| ‘Irand224()’
| ‘Uniform01()’

⟨*num-lit*⟩ ::= [‘+’ | ‘−’] ⟨*digits*⟩ [‘.’ ⟨*digits*⟩] [(‘d’ | ‘D’ | ‘e’ | ‘E’) [‘+’ | ‘−’] ⟨*digits*⟩]

⟨*sym-expr*⟩ ::= ⟨*cond-sym-expr*⟩ | ⟨*sym-expr-2*⟩

⟨*cond-sym-expr*⟩ ::= ‘if’ ⟨*logic-expr*⟩ ‘then’ ⟨*sym-expr*⟩ [‘else’ ⟨*sym-expr*⟩]

⟨*sym-expr-2*⟩ ::= ⟨*sym-expr-1*⟩ {‘&’ ⟨*sym-expr-1*⟩}

⟨*sym-expr-1*⟩ ::= ⟨*sym-ref*⟩ | ⟨*sym-func-ref*⟩ | ⟨*string-lit*⟩ | ⟨*num-expr*⟩ | ‘(’ ⟨*sym-expr*⟩ ‘)’ | ⟨*sym-expr*⟩

⟨*sym-ref*⟩ ::= ⟨*subscr-ref*⟩ | ⟨*unsubscr-ref*⟩

⟨*sym-func-ref*⟩ ::= ‘substr(’ ⟨*sym-expr*⟩ ‘,’ ⟨*num-expr*⟩ [‘,’ ⟨*num-expr*⟩] ‘)’
| ‘time2str(’ ⟨*num-expr*⟩ ‘,’ ⟨*sym-expr*⟩ ‘)’

⟨*string-lit*⟩ ::= ‘"’ ⟨*char-seq*⟩ ‘"’ | ‘'’ ⟨*char-seq*⟩ ‘'’

⟨*set-expr*⟩ ::= ⟨*cond-set-expr*⟩ | ⟨*set-expr-5*⟩

⟨*cond-set-expr*⟩ ::= ‘`if`’ ⟨*logic-expr*⟩ ‘`then`’ ⟨*set-expr-5*⟩ ‘`else`’ ⟨*set-expr-5*⟩

⟨*set-expr-5*⟩ ::= ⟨*set-expr-4*⟩ {(‘`union`’ | ‘`diff`’ | ‘`symdiff`’) ⟨*set-expr-4*⟩}

⟨*set-expr-4*⟩ ::= ⟨*set-expr-3*⟩ {‘`inter`’ ⟨*set-expr-3*⟩}

⟨*set-expr-3*⟩ ::= ⟨*set-expr-2*⟩ {‘`cross`’ ⟨*set-expr-2*⟩}

⟨*set-expr-2*⟩ ::= ⟨*iter-set-expr*⟩ | ⟨*arith-set*⟩ | ⟨*set-expr-1*⟩

⟨*iter-set-expr*⟩ ::= ‘`setof`’ ⟨*ind-expr*⟩ (⟨*tuple*⟩ | ⟨*simple-expr*⟩) | ⟨*ind-expr*⟩

⟨*arith-set*⟩ ::= ⟨*num-expr*⟩ ‘`..`’ ⟨*num-expr*⟩ [‘`by`’ ⟨*num-expr*⟩]

⟨*set-expr-1*⟩ ::= ⟨*set-ref*⟩ | ⟨*set-lit*⟩ | ‘`(`’ ⟨*set-expr*⟩ ‘`)`’ | ⟨*set-expr*⟩

⟨*set-ref*⟩ ::= ⟨*subscr-ref*⟩ | ⟨*unsubscr-ref*⟩

⟨*set-lit*⟩ ::= ‘`{`’ ‘`}`’
| ‘`{`’ ⟨*tuple*⟩ {‘`,`’ ⟨*tuple*⟩} ‘`}`’
| ‘`{`’ ⟨*simple-expr*⟩ {‘`,`’ ⟨*simple-expr*⟩} ‘`}`’

⟨*ind-expr*⟩ ::= ‘`{`’ ⟨*ind-entry*⟩ {‘`,`’ ⟨*ind-entry*⟩} [‘`:`’ ⟨*logic-expr*⟩] ‘`}`’

⟨*ind-entry*⟩ ::= ⟨*ind-entry-tuple*⟩ ‘`in`’ ⟨*set-expr*⟩
| ⟨*ident*⟩ ‘`in`’ ⟨*set-expr*⟩
| ⟨*set-expr*⟩

⟨*ind-entry-tuple*⟩ ::= ‘`(`’ (⟨*ident*⟩ | ⟨*simple-expr*⟩) {‘`,`’ (⟨*ident*⟩ | ⟨*simple-expr*⟩)
} ‘`)`’

⟨*logic-expr*⟩ ::= ⟨*logic-expr-5*⟩ {(‘`or`’ | ‘`||`’) ⟨*logic-expr-5*⟩}

⟨*logic-expr-5*⟩ ::= ⟨*iter-logic-expr*⟩ | ⟨*logic-expr-4*⟩

⟨*iter-logic-expr*⟩ ::= (‘`forall`’ | ‘`exists`’) ⟨*ind-expr*⟩ ⟨*logic-expr-4*⟩

⟨*logic-expr-4*⟩ ::= ⟨*logic-expr-3*⟩ {(‘`and`’ | ‘`&&`’) ⟨*logic-expr-3*⟩}

⟨*logic-expr-3*⟩ ::= (‘`not`’ | ‘`!`’) ⟨*logic-expr-2*⟩

⟨*logic-expr-2*⟩ ::= ⟨*rel-expr*⟩ | ⟨*logic-expr-1*⟩

⟨*rel-expr*⟩ ::= ⟨*simple-expr*⟩ (‘<’ | ‘<=’ | ‘>’ | ‘>=’) ⟨*simple-expr*⟩
| ⟨*simple-expr*⟩ (‘==’ | ‘=’ | ‘<>’ | ‘`!=`’) ⟨*simple-expr*⟩
| (⟨*tuple*⟩ | ⟨*simple-expr*⟩) [‘`not`’ | ‘`!`’] ‘`in`’ ⟨*set-expr*⟩
| ⟨*set-expr*⟩ [‘`not`’ | ‘`!`’] ‘`within`’ ⟨*set-expr*⟩

⟨*logic-expr-1*⟩ ::= ⟨*num-expr*⟩ | ‘(’ ⟨*logic-expr*⟩ ‘)’ | ⟨*logic-expr*⟩

⟨*lin-expr*⟩ ::= ⟨*cond-lin-expr*⟩ | ⟨*lin-expr-5*⟩

⟨*cond-lin-expr*⟩ ::= ‘`if`’ ⟨*logic-expr*⟩ ‘`then`’ ⟨*lin-expr-5*⟩ [‘`else`’ ⟨*lin-expr-5*⟩]

⟨*lin-expr-5*⟩ ::= ⟨*lin-expr-4*⟩ {(‘+’ | ‘−’) ⟨*lin-expr-4*⟩}

⟨*lin-expr-4*⟩ ::= ⟨*iter-lin-expr*⟩ | ⟨*lin-expr-3*⟩

⟨*iter-lin-expr*⟩ ::= ‘`sum`’ ⟨*ind-expr*⟩ ⟨*lin-expr-3*⟩

⟨*lin-expr-3*⟩ ::= ⟨*lin-expr-2*⟩ {(‘⋆’ | ‘/’) ⟨*num-expr-3*⟩}
| ⟨*num-expr-3*⟩ {‘⋆’ ⟨*lin-expr-2*⟩}

⟨*lin-expr-2*⟩ ::= ⟨*lin-expr-1*⟩ | (‘+’ | ‘−’) ⟨*lin-expr-1*⟩

⟨*lin-expr-1*⟩ ::= ⟨*lin-ref*⟩ | ⟨*num-func-ref*⟩ | ⟨*num-lit*⟩ | ‘(’ ⟨*lin-expr*⟩ ‘)’ | ⟨*lin-expr*⟩

⟨*lin-ref*⟩ ::= ⟨*subscr-ref*⟩ | ⟨*unsubscr-ref*⟩

⟨*ident*⟩ ::= ((⟨*alpha*⟩ | ‘_’) {⟨*alphanum*⟩ | ‘_’}) - ⟨*keyword*⟩

⟨*keyword*⟩ ::= ‘`if`’ | ‘`then`’ | ‘`else`’ | ‘`mod`’ | ‘`div`’ | ‘`less`’ | ‘`within`’
| ‘`union`’ | ‘`cross`’ | ‘`diff`’ | ‘`inter`’ | ‘`symdiff`’ | ‘`in`’ | ‘`by`’
| ‘`and`’ | ‘`or`’ | ‘`not`’

⟨*alphanum*⟩ ::= (⟨*letter*⟩ | ⟨*digit*⟩) {(⟨*letter*⟩ | ⟨*digit*⟩)}

⟨*alpha*⟩ ::= ⟨*letter*⟩ {⟨*letter*⟩}

⟨*letter*⟩ ::= ‘`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`’ | ‘`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`’

⟨*digits*⟩ ::= ⟨*digit*⟩ {⟨*digit*⟩}

⟨*digit*⟩ ::= ‘`0`’ | ‘`1`’ | ‘`2`’ | ‘`3`’ | ‘`4`’ | ‘`5`’ | ‘`6`’ | ‘`7`’ | ‘`8`’ | ‘`9`’

⟨*char-seq*⟩ ::= (* Arbitrary character sequence. Characters used as delimiters (single or double quotes) can only be part of the sequence if they are coded twice. *)