Context Free Grammar for the mDraw Gcode syntax
(To aid in designing the parser)

anything wrapped inside parentheses () is a nonterminal: they have further syntax rules. Anything not wrapped in parentheses is a terminal, they have no further syntax rules and should be considered concrete characters or string of characters, **a space character <Sp>**, **a linefeed character (<LF>, or an empty character <e> - meaning that there is nothing there to parse anymore for that particular nonterminal**).

```
GCODE                -> M(MCODE)<LF> | G(GCODE)<LF>

(MCODE)              -> 10(COMOPEN)                    |
                        11(LIMIT-SW-QUERY)             |
                        1<Sp>(SET-PEN-POS)             |
                        2<Sp>(SAVE-PEN-UD-POS)         |
                        4<Sp>(SET-LASER-POW)           |
                        5<Sp>(SAVE-STEPPER-INFO)

(COMOPEN)            -> <e>
(LIMIT-SW-QUERY)     -> <e>
(SET-PEN-POS)        -> (8-BIT-NUMBER)
(SAVE-PEN-UD-POS)    -> U(8-BIT-NUMBER)<Sp>D(8-BIT-NUMBER)
(SET-LASER-POW)      -> (8-BIT-NUMBER)
(SAVE-STEPPER-INFO)-> A(DIR)<Sp>B(DIR)<Sp>H(NUMBER)
                        <Sp>W(NUMBER)<Sp>S(NUMBER)

(GCODE)              -> 28(GOTO-ORIGIN) | 1<Sp>(GOTO-POSITION)

(GOTO-ORIGIN)        -> <e>
(GOTO-POSITION)      -> X(NUMBER).(NUMBER)<Sp>Y(NUMBER).(NUMBER)
                       <Sp>(RELATIVITY-MODE)

(RELATIVITY-MODE)    -> A0 | A1

(NUMBER)             -> (DIGIT)(NUMBER) | (DIGIT)

(8-BIT-NUMBER)       -> (8-BIT-HUNDREDS)(DIGIT)(DIGIT) |
                        (DIGIT)(DIGIT) | (DIGIT)
(DIR)                -> 0 | 1
(8-BIT-HUNDREDS)     -> 0 | 1 | 2
(DIGIT)              -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Further reflections on how to design the actual parser:

- It might be a good idea to tokenize the input string first.

- The code number (after M or G) could be transformed into an int and then checked which code number it matches and a proper parser function could be called based on the matched code number (or an error message sent, if no such code number has been found):

```
switch (codenum) {
     case '1':
          setPenPos(rest-of-tokens);
          break;
     case '2':
          savePenUdPos(rest-of-tokens);
          break;
     // …
     default:
          //handle erroneous code number here
}
```

  where rest-of-tokens is an array pointer to the rest of the “list” of input tokens