

1. Introduction

1.1. This language reference manual lays out the specifications for a programming language which can be used to prototype and analyze knitting patterns.

2. Lexical conventions

2.1. Comments begin with the reserved keyword **Note :** and continue until the next newline character.

2.2. Whitespace is ignored, except as necessary to delineate comments and tokens.

2.3. Keywords and phrases are case sensitive. Reserved keywords may not be used as identifiers.

B0

C0

count

Chart

Dec

EndStitchPattern

inc

k

p

Pattern

remain

Row

StitchPattern

times

total

until

Work

- 2.4. Identifiers begin with a letter and may be followed by a series of letters, numbers, and underscores ('_'). Identifiers are not allowed to consist of only the letters 'k' or 'p' followed by numbers, because that syntax is reserved for knit and purl stitch notation.
- 2.5. Literals consist of a series of ASCII digits, representing a base-10 integer.
- 2.6. The symbols ':', '(', ')', ';', ',', '*' and '=' are used as part of program syntax. Use of other symbols is a syntax error.

3. Program structure

- 3.1. A program consists of an optional series of row and stitch pattern declarations, a main pattern, and an optional series of queries to perform on the completed pattern. Row and stitch pattern declarations may be interspersed, but must be declared before they can be invoked.
- 3.2. A row declaration consists of the keyword "Row", the row name identifier, an optional list of formal parameter identifiers in parentheses, a colon, an optional list of beginning stitches to work, an optional list of repeated stitches, set off by stitch-repeat markers('*') and followed by a repeat condition clause, and finally an optional list of ending stitches to work. At least one of the beginning stitch list, repeated stitch clause, or ending stitch list must be non-empty.
 - 3.2.1. Parameters may consist of either literals or identifiers. Any non-literal parameters used within a row declaration must be identifiers included in the list of formal parameter identifiers.
 - 3.2.2. Lists of stitches are comma-separated sequences of the keywords 'k', 'p', 'inc' and 'dec'. Sequences of consecutive 'k' and 'p' stitches may be condensed into 'k/p <PARAM>' format, so that 'p 4' is equivalent to 'p,p,p,p'.
 - 3.2.3. A repeat condition clause has one of two formats:
 - 3.2.3.1. "<PARAM> times" which explicitly specifies the number of times to repeat the enclosed stitches.
 - 3.2.3.2. "until <PARAM> remain" which causes the enclosed stitches to be repeated until <PARAM> stitches remain unworked in the row.
 - 3.2.4. Example row declaration:

```
Row increase (amtToIncrease): k1,p1 *inc* amtToIncrease times p1,k1
```
- 3.3. Stitch pattern declarations consist of the keyword "StitchPattern", the stitch pattern name identifier, an optional list of formal parameter identifiers in parentheses, a colon, a semicolon separated series of row instructions, and the "EndStichPattern" keyword.

3.3.1. Row instructions have two formats:

3.3.1.1. “Work Row <ID> (<Param>, <Param>, ...)” where a previously declared row identifier name and parenthesized list of actual parameter values means to work the specified row with the given parameter values as input.

3.3.1.2. An inlined version of the row declaration, with the same format as above, followed by a parenthesized list of actual parameter values.

3.3.2. In both cases, any identifiers used as actual parameter values must have been declared as formal parameters to the stitch pattern, and the provided lists of actual parameter values must correspond in length to the declared lists of formal parameter identifiers.

3.3.3. Example stitch pattern declaration:

```
StitchPattern foo (a) :  
    Row x: k2 *k1* a times k2;  
    Work Row increase (a);  
    Row z: k a *p* until a remain k a  
EndStitchPattern
```

3.4. The main pattern definition consists of the keyword “Pattern”, a cast-on statement, a semicolon separated series of pattern instructions, and the bind-off keyword “BO”.

3.4.1. The cast-on statement consists of the keyword “CO” and an integer literal representing the initial number of stitches in a row.

3.4.1.1. Ex: “CO 42”

3.4.2. A pattern instruction has one of the following formats:

3.4.2.1. “<ID> = <Literal>” which assigns the literal value to the identifier and makes the identifier available to be used in actual parameter value lists.

3.4.2.2. “<ID> = count” which assigns the number of stitches in the current row to the given identifier, and makes the identifier available to be used in actual parameter value lists.

3.4.2.3. A row instruction, as defined for stitch patterns

3.4.2.4. “Work StitchPattern <ID> (<Param>, <Param>, ...)” where a previously declared stitch pattern identifier name and parenthesized list of actual parameter values means to work all rows of the specified stitch pattern once with the given parameter values as input.

3.4.2.5. The pattern instruction repeat symbol “**”, a semicolon separated list of non-repeat-containing pattern instructions, the instruction repeat symbol, and a repeat condition in one of two forms:

3.4.2.5.1. “<PARAM> times” which explicitly specifies the number of times to repeat the enclosed instructions.

3.4.2.5.2. “until <PARAM> total” which causes the enclosed instructions to be repeated until the working row has <PARAM> stitches.

3.5. Following the complete pattern declaration, three output queries are supported.

3.5.1. “count” outputs the number of rows in the pattern from cast-on to bind-off as an integer to stdout.

3.5.2. “count(<Literal>)” outputs the number of stitches in the numbered row of the pattern.

3.5.3. “chart” outputs a full pattern schematic representing all of the stitches in the pattern. For example:

