

*Copyright © 1994 by Sean Luke*

**COWS Interpreter**

# **Parser Grammar**

**COWS Version 1.3**

**Sean Luke**

**March 20, 1994**

This file describes the grammar used by the COWS Interpreter when it parses through code, breaking the code into functions and global variables. It should give you a peek into how the parser's private methods work.

# Special Functions

The following methods do special things along with parsing:

- ◆ *variablelist* loads globals into `global_dictionary`.
- ◆ *localvariablelist* loads variables into a local dictionary.
- ◆ *argumentlist* loads arguments into a local argument list.
- ◆ *itemlist* grabs the remainder of the function.
- ◆ *functionform* loads arguments from the argument list into the local dictionary and puts the local dictionary and item list into `function_dictionary`.

# Non-Terminals

program            :-    <programlist>

```

programlist      :- EOF | <globalform> <programlist> | <functionform>
                  <programlist>
globalform       :- ( <globalkeyword> <variablelist> )
functionform     :- (function <functionname> <argumentlist>
                    variable <localvariablelist> do <itemlist> |
                    (function <functionname> <argumentlist>
                     do <itemlist>
variablelist     :- <symbolname> <variablelist> | NULL
localvariablelist :- <symbolname> <localvariablelist> | NULL
argumentlist     :- <symbolname> <argumentlist> | NULL
itemlist         :-<item>* )  note I use a while-loop here. Less recursive.
item             :-<atom> | ( <itemlist>

```

# Terminals

- ◆ *globalkeyword* is the word "variable".
- ◆ *functionkeyword* is the word "function".
- ◆ *symbolname* is any non-string, non-truth, non-number, non-keyword, non-delimiter.
- ◆ *atom* is any non-delimiter.