

BNF for the Gryph Programming Language

Vitor Greati

Artur Curinga

Carlos Vieira

Vinícius Campos

May 18, 2018

Contents

1	General structure	1
1.1	Program	1
1.2	Statements	1
1.2.1	IO	1
1.2.2	Variables	2
1.3	Subprograms	2
1.3.1	Declaration	2
1.3.2	Call	2
2	Control Structures	2
2.1	If-else statements	2
3	Types	2
4	Expressions	2
4.1	Literals	3

1 General structure

1.1 Program

$$\begin{aligned}\langle \text{program} \rangle &\models \langle \text{program-unit} \rangle \mid \langle \text{program-unit} \rangle \langle \text{program} \rangle \\ \langle \text{program-unit} \rangle &\models \langle \text{stmt} \rangle ; \mid \langle \text{subprog-decl} \rangle\end{aligned}$$

1.2 Statements

$$\begin{aligned}\langle \text{stmt-list} \rangle &\models \langle \text{stmt} \rangle ; \mid \langle \text{stmt} \rangle ; \langle \text{stmt-list} \rangle \\ \langle \text{stmt-block} \rangle &\models \{ \langle \text{stmt-list} \rangle \} \\ \langle \text{block-or-matched} \rangle &\models \langle \text{stmt-block} \rangle \mid \langle \text{matched-stmt} \rangle ; \\ \langle \text{com-stmt} \rangle &\models \langle \text{read-stmt} \rangle \mid \langle \text{print-stmt} \rangle \mid \langle \text{var-decl-stmt} \rangle \\ \langle \text{stmt} \rangle &\models \langle \text{matched-stmt} \rangle \mid \langle \text{unmatched-stmt} \rangle \\ \langle \text{matched-stmt} \rangle &\models \langle \text{matched-if-else} \rangle \mid \langle \text{com-stmt} \rangle \\ \langle \text{unmatched-stmt} \rangle &\models \langle \text{if-stmt} \rangle \mid \langle \text{unmatched-if-else} \rangle\end{aligned}$$

1.2.1 IO

$$\begin{aligned}\langle \text{read-stmt} \rangle &\models \text{read } \langle \text{ident} \rangle \\ \langle \text{write-stmt} \rangle &\models \text{print } \langle \text{ident} \rangle \mid \text{print } \langle \text{string-lit} \rangle\end{aligned}$$

1.2.2 Variables

$$\begin{aligned}\langle \text{ident-begin-stmt} \rangle &\models \langle \text{ident-list} \rangle \langle \text{ident-list-post} \rangle \\ \langle \text{ident-list-post} \rangle &\models :\langle \text{type} \rangle \langle \text{var-decl-stmt} \rangle \mid \langle \text{var-attr-stmt} \rangle \\ \langle \text{var-decl-stmt} \rangle &\models \lambda \mid \langle \text{var-attr-stmt} \rangle \\ \langle \text{var-attr-stmt} \rangle &\models =\langle \text{expr-list} \rangle\end{aligned}$$

1.3 Subprograms

1.3.1 Declaration

1.3.2 Call

$$\langle \text{subprog-call} \rangle \models \langle \text{ident} \rangle (\langle \text{expr-list} \rangle)$$

2 Control Structures

2.1 If-else statements

$$\begin{aligned}\langle \text{if-expr} \rangle &\models \text{if } (\langle \text{b-exp} \rangle) \\ \langle \text{if-stmt} \rangle &\models \langle \text{if-expr} \rangle \langle \text{stmt} \rangle; \\ \langle \text{unmatched-if-else} \rangle &\models \langle \text{if-expr} \rangle \langle \text{matched-stmt} \rangle; \text{else } \langle \text{unmatched-stmt} \rangle; \\ \langle \text{matched-if-else} \rangle &\models \langle \text{if-expr} \rangle \langle \text{block-or-matched} \rangle \text{else } \langle \text{block-or-matched} \rangle \mid \langle \text{if-expr} \rangle \langle \text{stmt-block} \rangle\end{aligned}$$

3 Types

$$\begin{aligned}\langle \text{type-list} \rangle &\models \langle \text{type} \rangle, \langle \text{type-list} \rangle \mid \langle \text{type} \rangle \\ \langle \text{type} \rangle &\models \langle \text{native-type} \rangle \mid \langle \text{user-type} \rangle \\ \langle \text{native-type} \rangle &\models \langle \text{primitive-type} \rangle \mid \langle \text{composite-type} \rangle \\ \langle \text{primitive-type} \rangle &\models \text{int} \mid \text{float} \mid \text{char} \mid \text{string} \\ \langle \text{composite-type} \rangle &\models [\langle \text{type} \rangle] \mid |\langle \text{type} \rangle| \mid (\langle \text{type} \rangle, \langle \text{type-list} \rangle) \mid \langle \text{graph-type} \rangle \\ \langle \text{graph-type} \rangle &\models <\langle \text{type} \rangle> \mid <\langle \text{type} \rangle, \langle \text{type} \rangle> \\ \langle \text{user-type} \rangle &\models \langle \text{upper-letter} \rangle \langle \text{alpha-num-list} \rangle\end{aligned}$$

Observations

- The maximum size of tuples depends on the language implementation, though, in the BNF description above, it may assume any value.

4 Expressions

$$\begin{aligned}\langle \text{expression} \rangle &\models \langle \text{logical-xor-expr} \rangle \\ \langle \text{logical-xor-expr} \rangle &\models \langle \text{logical-or-expr} \rangle \mid \langle \text{logical-or-expr} \rangle \langle \text{logical-xor-expr-aux} \rangle \\ \langle \text{logical-xor-expr-aux} \rangle &\models \text{xor } \langle \text{logical-or-expr} \rangle \mid \text{xor } \langle \text{logical-or-expr} \rangle \langle \text{logical-xor-expr-aux} \rangle \\ \langle \text{logical-or-expr} \rangle &\models \langle \text{logical-and-expr} \rangle \mid \langle \text{logical-and-expr} \rangle \langle \text{logical-or-expr-aux} \rangle \\ \langle \text{logical-or-expr-aux} \rangle &\models \text{or } \langle \text{logical-and-expr} \rangle \mid \text{or } \langle \text{logical-and-expr} \rangle \langle \text{logical-or-expr-aux} \rangle \\ \langle \text{logical-and-expr} \rangle &\models \langle \text{equality-expr} \rangle \mid \langle \text{equality-expr} \rangle \langle \text{logical-and-expr-aux} \rangle\end{aligned}$$

$\langle \text{logical-and-expr-aux} \rangle$	\models	$\text{and } \langle \text{equality-expr} \rangle \mid \text{and } \langle \text{equality-expr} \rangle \langle \text{logical-and-expr-aux} \rangle$
$\langle \text{equality-expr} \rangle$	\models	$\langle \text{rel-expr} \rangle \mid \langle \text{rel-expr} \rangle \langle \text{rel-expr-aux} \rangle$
$\langle \text{equality-expr-aux} \rangle$	\models	$\langle \text{equality-op} \rangle \langle \text{rel-expr} \rangle \mid \langle \text{equality-op} \rangle \langle \text{rel-expr} \rangle \langle \text{equality-expr-aux} \rangle$
$\langle \text{rel-expr} \rangle$	\models	$\langle \text{add-expr} \rangle \langle \text{rel-expr-aux} \rangle$
$\langle \text{rel-expr-aux} \rangle$	\models	$\langle \text{rel-op} \rangle \langle \text{add-expr} \rangle \mid \langle \text{rel-op} \rangle \langle \text{add-expr} \rangle \langle \text{rel-expr-aux} \rangle$
$\langle \text{add-expr} \rangle$	\models	$\langle \text{mult-expr} \rangle \mid \langle \text{mult-expr} \rangle \langle \text{add-expr-aux} \rangle$
$\langle \text{add-expr-aux} \rangle$	\models	$\langle \text{add-op} \rangle \langle \text{mult-expr} \rangle \mid \langle \text{add-op} \rangle \langle \text{mult-expr} \rangle \langle \text{add-expr-aux} \rangle$
$\langle \text{mult-expr} \rangle$	\models	$\langle \text{exp-expr} \rangle \mid \langle \text{exp-expr} \rangle \langle \text{mult-expr-aux} \rangle$
$\langle \text{mult-expr-aux} \rangle$	\models	$\langle \text{mult-op} \rangle \langle \text{exp-expr} \rangle \mid \langle \text{mult-op} \rangle \langle \text{exp-expr} \rangle \langle \text{mult-expr-aux} \rangle$
$\langle \text{exp-expr} \rangle$	\models	$\langle \text{cast-expr} \rangle \mid \langle \text{cast-expr} \rangle \langle \text{exp-op} \rangle \langle \text{exp-expr} \rangle$
$\langle \text{cast-expr} \rangle$	\models	$\langle \text{unary-expr} \rangle \mid \langle \text{unary-expr} \rangle \langle \text{cast-expr-aux} \rangle$
$\langle \text{cast-expr-aux} \rangle$	\models	$@ \langle \text{type} \rangle \mid @ \langle \text{type} \rangle \langle \text{cast-expr-aux} \rangle$
$\langle \text{unary-expr} \rangle$	\models	$\langle \text{unary-op} \rangle \langle \text{cast-expr} \rangle \mid \langle \text{postfix-expr} \rangle$
$\langle \text{postfix-expr} \rangle$	\models	$\langle \text{primary-expr} \rangle \mid \langle \text{ident} \rangle \mid \text{expression} \mid \langle \text{ident} \rangle \langle \text{expression} \rangle \mid \langle \text{ident} \rangle [\text{expression}] \mid \langle \text{ident} \rangle \{ \langle \text{ident} \rangle \} \mid \langle \text{ident} \rangle . \langle \text{expression} \rangle$
$\langle \text{primary-expr} \rangle$	\models	$\langle \text{expression} \rangle \mid \langle \text{ident} \rangle \mid \langle \text{subprogcall} \rangle \mid \langle \text{constant} \rangle$
$\langle \text{constant} \rangle$	\models	$\langle \text{int-lit} \rangle \mid \langle \text{float-lit} \rangle \mid \langle \text{string-lit} \rangle \mid \langle \text{bool-lit} \rangle \mid \langle \text{list-lit} \rangle \mid \langle \text{graph-lit} \rangle$
$\langle \text{rel-op} \rangle$	\models	$> \mid < \mid < = \mid > =$
$\langle \text{equality-op} \rangle$	\models	$= = \mid ! =$
$\langle \text{unary-op} \rangle$	\models	$+ \mid -$
$\langle \text{add-op} \rangle$	\models	$+ \mid -$
$\langle \text{mult-op} \rangle$	\models	$* \mid / \mid \% \mid ++ \mid **$
$\langle \text{exp-op} \rangle$	\models	$^$

4.1 Literals

$\langle \text{list-lit} \rangle$	\models	$[\langle \text{expression-list} \rangle]$
$\langle \text{tuple-lit} \rangle$	\models	$(\langle \text{expression-list} \rangle)$
$\langle \text{dict-entry} \rangle$	\models	$\langle \text{expression} \rangle ? \langle \text{expression} \rangle$
$\langle \text{dict-entry-list} \rangle$	\models	$\langle \text{dict-entry} \rangle \mid \langle \text{dict-entry} \rangle , \langle \text{dict-entry-list} \rangle$
$\langle \text{dict-lit} \rangle$	\models	$\{ \langle \text{dict-entry-list} \rangle \}$