BNF for the Gryph Programming Language

Vitor Greati   Artur Curinga   Carlos Vieira   Vinícius Campos

May 17, 2018

Contents

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

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

⟨logical-and-expr-aux⟩	⊨	and ⟨equality-expr⟩ | and ⟨equality-expr⟩⟨logical-and-expr-aux⟩
⟨equality-expr⟩	⊨	⟨rel-expr⟩ | ⟨rel-expr⟩⟨rel-expr-aux⟩
⟨equality-expr-aux⟩	⊨	⟨equality-op⟩ ⟨rel-expr⟩ | ⟨equality-op⟩ ⟨rel-expr⟩⟨equality-expr-aux⟩
⟨rel-expr⟩	⊨	⟨add-expr⟩⟨rel-expr-aux⟩
⟨rel-expr-aux⟩	⊨	⟨rel-op⟩⟨add-expr⟩ | ⟨rel-op⟩⟨add-expr⟩⟨rel-expr-aux⟩
⟨add-expr⟩	⊨	⟨mult-expr⟩ | ⟨mult-expr⟩⟨add-expr-aux⟩
⟨add-expr-aux⟩	⊨	⟨add-op⟩⟨mult-expr⟩ | ⟨add-op⟩⟨mult-expr⟩⟨add-expr-aux⟩
⟨mult-expr⟩	⊨	⟨exp-expr⟩ | ⟨exp-expr⟩⟨mult-expr-aux⟩
⟨mult-expr-aux⟩	⊨	⟨mult-op⟩⟨exp-expr⟩ | ⟨mult-op⟩⟨exp-expr⟩⟨mult-expr-aux⟩
⟨exp-expr⟩	⊨	⟨cast-expr⟩ | ⟨cast-expr⟩⟨exp-op⟩⟨exp-expr⟩
⟨cast-expr⟩	⊨	⟨unary-expr⟩ | ⟨unary-expr⟩⟨cast-expr-aux⟩
⟨cast-expr-aux⟩	⊨	@⟨type⟩ | @⟨type⟩⟨cast-expr-aux⟩
⟨unary-expr⟩	⊨	⟨unary-op⟩⟨cast-expr⟩ | ⟨postfix-expr⟩
⟨postfix-expr⟩	⊨	⟨primary-expr⟩ | ⟨ident⟩|expression| | ⟨ident⟩<expression> | ⟨ident⟩[expression] | ⟨ident⟩{⟨ident⟩} | ⟨ident⟩.⟨expression⟩
⟨primary-expr⟩	⊨	(⟨expression⟩) | ⟨ident⟩ | ⟨subprogcall⟩ | ⟨constant⟩
⟨constant⟩	⊨	⟨int-lit⟩ | ⟨float-lit⟩ | ⟨string-lit⟩ | ⟨bool-lit⟩ | ⟨list-lit⟩ | ⟨graph-lit⟩
⟨rel-op⟩	⊨	> | < | <= | >=
⟨equality-op⟩	⊨	== | !=
⟨unary-op⟩	⊨	+ | -
⟨add-op⟩	⊨	+ | -
⟨mult-op⟩	⊨	* | / | % | ++ | **
⟨exp-op⟩	⊨	ˆ