

HoCL v1.2 Syntax

J. Sérot



$\langle \text{program} \rangle ::= \langle \text{top_decl} \rangle^* \text{ EOF}$
 $\langle \text{top_decl} \rangle ::= \begin{array}{l} \langle \text{type_decl} \rangle ; \\ | \\ \langle \text{val_decl} \rangle ; \\ | \\ \langle \text{node_decl} \rangle ; \end{array}$
 $\langle \text{type_decl} \rangle ::= \text{type IDENT}$
 $\langle \text{node_decl} \rangle ::= \begin{array}{l} \text{node IDENT in } \langle \text{io_decls} \rangle \text{ out } \langle \text{io_decls} \rangle [\langle \text{node_impl} \rangle] \\ | \\ \text{graph IDENT in } \langle \text{io_decls} \rangle \text{ out } \langle \text{io_decls} \rangle \langle \text{node_impl} \rangle \end{array}$
 $\langle \text{node_impl} \rangle ::= \begin{array}{l} \text{fun } \langle \text{val_decl} \rangle^* \text{ end} \\ | \\ \text{struct } \langle \text{struct_graph_desc} \rangle \text{ end} \\ | \\ \text{actor } \langle \text{actor_desc} \rangle^* \text{ end} \end{array}$
 $\langle \text{actor_desc} \rangle ::= \text{IDENT (} \langle \text{impl_attr} \rangle^* \text{)}$
 $\langle \text{impl_attr} \rangle ::= \begin{array}{l} \text{IDENT = STRING} \\ | \\ \text{IDENT} \end{array}$
 $\langle \text{io_decls} \rangle ::= (\langle \text{io_decl} \rangle^*)$
 $\langle \text{io_decl} \rangle ::= \text{IDENT : } \langle \text{type_expr} \rangle [\langle \text{io_expr} \rangle] \langle \text{io_annots} \rangle$
 $\langle \text{io_expr} \rangle ::= = \langle \text{simple_expr} \rangle$
 $\langle \text{io_annots} \rangle ::= \begin{array}{l} \epsilon \\ | \\ [\langle \text{basic_expr} \rangle] \\ | \\ \{ \langle \text{io_annot} \rangle^* \} \end{array}$
 $\langle \text{io_annot} \rangle ::= \text{IDENT = STRING}$
 $\langle \text{simple_type_expr} \rangle ::= \begin{array}{l} \text{IDENT} \\ | \\ \text{TYVAR} \end{array}$
 $\langle \text{type_expr} \rangle ::= \begin{array}{l} \langle \text{simple_type_expr} \rangle \\ | \\ \langle \text{simple_type_expr} \rangle \text{ IDENT} \end{array}$
 $\langle \text{val_decl} \rangle ::= \text{val [rec] } \langle \text{binding} \rangle_{\text{and}}^+$
 $\langle \text{binding} \rangle ::= \begin{array}{l} \langle \text{pattern} \rangle = \langle \text{expr} \rangle \\ | \\ \langle \text{binding_name} \rangle \langle \text{fun_pattern} \rangle^+ = \langle \text{expr} \rangle \end{array}$
 $\langle \text{binding_name} \rangle ::= \begin{array}{l} \text{IDENT} \\ | \\ (\text{INFIX0}) \end{array}$
 $\langle \text{expr} \rangle ::= \begin{array}{l} \langle \text{simple_expr} \rangle \\ | \\ \langle \text{simple_expr} \rangle \langle \text{simple_expr} \rangle^+ \\ | \\ \langle \text{expr_comma_list} \rangle \\ | \\ \text{fun } \langle \text{fun_pattern} \rangle^+ \rightarrow \langle \text{expr} \rangle \\ | \\ \text{let [rec] } \langle \text{binding} \rangle_{\text{and}}^+ \text{ in } \langle \text{expr} \rangle \\ | \\ \text{if } \langle \text{expr} \rangle \text{ then } \langle \text{expr} \rangle \text{ else } \langle \text{expr} \rangle \\ | \\ \langle \text{expr} \rangle \text{ INFIX0 } \langle \text{expr} \rangle \end{array}$

		$\langle \text{expr} \rangle$ INFIX1 $\langle \text{expr} \rangle$
		$\langle \text{expr} \rangle$ INFIX2 $\langle \text{expr} \rangle$
		$\langle \text{expr} \rangle$ INFIX3 $\langle \text{expr} \rangle$
		$\langle \text{expr} \rangle = \langle \text{expr} \rangle$
		$\langle \text{expr} \rangle :: \langle \text{expr} \rangle$
		$\langle \text{simple_expr} \rangle$ [$\langle \text{simple_expr} \rangle$]
		match $\langle \text{expr} \rangle$ with $\langle \text{match_case} \rangle_{\text{BAR}}^+$
$\langle \text{match_case} \rangle$::=	$\langle \text{pattern} \rangle \rightarrow \langle \text{expr} \rangle$
$\langle \text{simple_expr} \rangle$::=	IDENT
		($\langle \text{expr} \rangle$)
		()
		$\langle \text{const_expr} \rangle$
		[$\langle \text{expr} \rangle^+$]
		[]
		' $\langle \text{basic_expr} \rangle$ '
$\langle \text{const_expr} \rangle$::=	INT
		true
		false
$\langle \text{expr_comma_list} \rangle$::=	$\langle \text{expr_comma_list} \rangle$, $\langle \text{expr} \rangle$
		$\langle \text{expr} \rangle$, $\langle \text{expr} \rangle$
$\langle \text{pattern} \rangle$::=	$\langle \text{simple_pattern} \rangle$
		$\langle \text{pattern_comma_list} \rangle$
		$\langle \text{pattern} \rangle :: \langle \text{pattern} \rangle$
		[$\langle \text{simple_pattern} \rangle^+$]
$\langle \text{fun_pattern} \rangle$::=	IDENT
$\langle \text{simple_pattern} \rangle$::=	IDENT
		-
		($\langle \text{pattern} \rangle$)
		()
		[]
$\langle \text{pattern_comma_list} \rangle$::=	$\langle \text{pattern_comma_list} \rangle$, $\langle \text{pattern} \rangle$
		$\langle \text{pattern} \rangle$, $\langle \text{pattern} \rangle$
$\langle \text{struct_graph_desc} \rangle$::=	$\langle \text{struct_decl} \rangle^*$
$\langle \text{struct_decl} \rangle$::=	$\langle \text{wire_decl} \rangle$
		$\langle \text{box_decl} \rangle$
$\langle \text{wire_decl} \rangle$::=	wire IDENT* : $\langle \text{type_expr} \rangle$
$\langle \text{box_decl} \rangle$::=	box IDENT : IDENT $\langle \text{box_inps} \rangle$ $\langle \text{box_outps} \rangle$
$\langle \text{box_inps} \rangle$::=	($\langle \text{box_inp} \rangle^*$)
$\langle \text{box_outps} \rangle$::=	($\langle \text{box_outp} \rangle^*$)

$$\begin{aligned}
\langle \text{box_inp} \rangle &::= \text{IDENT} \\
&\quad | \quad ' \langle \text{basic_expr} \rangle ' \\
\langle \text{box_outp} \rangle &::= \text{IDENT} \\
\langle \text{basic_expr} \rangle &::= \text{IDENT} \\
&\quad | \quad \langle \text{const_expr} \rangle \\
&\quad | \quad \langle \text{basic_expr} \rangle \text{ INFIX1 } \langle \text{basic_expr} \rangle \\
&\quad | \quad \langle \text{basic_expr} \rangle \text{ INFIX2 } \langle \text{basic_expr} \rangle \\
&\quad | \quad \langle \text{basic_expr} \rangle \text{ INFIX3 } \langle \text{basic_expr} \rangle \\
&\quad | \quad (\langle \text{basic_expr} \rangle)
\end{aligned}$$

Lexical Syntax

$$\begin{aligned}
\text{IDENT} &::= \langle \text{letter} \rangle (\langle \text{letter} \rangle \mid \langle \text{digit} \rangle)^* \\
\langle \text{letter} \rangle &::= 'a', \dots, 'z', 'A', \dots, 'Z' \\
\text{INT} &::= [-] \langle \text{digit} \rangle^+ \\
\langle \text{digit} \rangle &::= '0', \dots, '9' \\
\text{INFIX1} &::= '=' \mid '!=' \mid '<' \mid '>' \\
\text{INFIX2} &::= '+' \mid '-' \\
\text{INFIX3} &::= '*' \mid '/' \mid '%' \\
\text{TYVAR} &::= '$' \langle \text{letter} \rangle
\end{aligned}$$