

# HoCL v1.2 Syntax

J. Sérot



$\langle \text{program} \rangle ::= \langle \text{top\_decl} \rangle^* \text{ EOF}$   
 $\langle \text{top\_decl} \rangle ::= \langle \text{type\_decl} \rangle ;$   
 $\quad \quad \quad | \quad \langle \text{val\_decl} \rangle ;$   
 $\quad \quad \quad | \quad \langle \text{node\_decl} \rangle ;$   
 $\langle \text{type\_decl} \rangle ::= \text{type IDENT}$   
 $\langle \text{node\_decl} \rangle ::= \text{node IDENT in } \langle \text{io\_decls} \rangle \text{ out } \langle \text{io\_decls} \rangle [\langle \text{node\_impl} \rangle]$   
 $\quad \quad \quad | \quad \text{graph IDENT in } \langle \text{io\_decls} \rangle \text{ out } \langle \text{io\_decls} \rangle \langle \text{node\_impl} \rangle$   
 $\langle \text{node\_impl} \rangle ::= \text{fun } \langle \text{val\_decl} \rangle^* \text{ end}$   
 $\quad \quad \quad | \quad \text{struct } \langle \text{struct\_graph\_desc} \rangle \text{ end}$   
 $\quad \quad \quad | \quad \text{actor } \langle \text{actor\_desc} \rangle^* \text{ end}$   
 $\langle \text{actor\_desc} \rangle ::= \text{IDENT ( } \langle \text{impl\_attr} \rangle^* \text{ )}$   
 $\langle \text{impl\_attr} \rangle ::= \text{IDENT = STRING}$   
 $\quad \quad \quad | \quad \text{IDENT}$   
 $\langle \text{io\_decls} \rangle ::= ( \langle \text{io\_decl} \rangle^* )$   
 $\langle \text{io\_decl} \rangle ::= \text{IDENT : } \langle \text{type\_expr} \rangle [\langle \text{io\_expr} \rangle] \langle \text{io\_annots} \rangle$   
 $\langle \text{io\_expr} \rangle ::= = \langle \text{simple\_expr} \rangle$   
 $\langle \text{io\_annots} \rangle ::= \epsilon$   
 $\quad \quad \quad | \quad [ \langle \text{basic\_expr} \rangle ]$   
 $\quad \quad \quad | \quad \{ \langle \text{io\_annot} \rangle^* , \}$   
 $\langle \text{io\_annot} \rangle ::= \text{IDENT = STRING}$   
 $\langle \text{simple\_type\_expr} \rangle ::= \text{IDENT}$   
 $\quad \quad \quad | \quad \text{TYVAR}$   
 $\langle \text{type\_expr} \rangle ::= \langle \text{simple\_type\_expr} \rangle$   
 $\quad \quad \quad | \quad \langle \text{simple\_type\_expr} \rangle \text{ IDENT}$   
 $\langle \text{val\_decl} \rangle ::= \text{val [rec] } \langle \text{binding} \rangle_{\text{and}}^+$   
 $\langle \text{binding} \rangle ::= \langle \text{pattern} \rangle = \langle \text{expr} \rangle$   
 $\quad \quad \quad | \quad \langle \text{binding\_name} \rangle \langle \text{fun\_pattern} \rangle^+ = \langle \text{expr} \rangle$   
 $\langle \text{binding\_name} \rangle ::= \text{IDENT}$   
 $\quad \quad \quad | \quad ( \text{ INFIX0 } )$   
 $\langle \text{expr} \rangle ::= \langle \text{simple\_expr} \rangle$   
 $\quad \quad \quad | \quad \langle \text{simple\_expr} \rangle \langle \text{simple\_labeled\_expr} \rangle^+$   
 $\quad \quad \quad | \quad \langle \text{expr\_comma\_list} \rangle$   
 $\quad \quad \quad | \quad \text{fun } \langle \text{fun\_pattern} \rangle^+ \rightarrow \langle \text{expr} \rangle$   
 $\quad \quad \quad | \quad \text{let [rec] } \langle \text{binding} \rangle_{\text{and}}^+ \text{ in } \langle \text{expr} \rangle$   
 $\quad \quad \quad | \quad \text{if } \langle \text{expr} \rangle \text{ then } \langle \text{expr} \rangle \text{ else } \langle \text{expr} \rangle$   
 $\quad \quad \quad | \quad \langle \text{expr} \rangle \text{ INFIX0 } \langle \text{expr} \rangle$

		$\langle \text{expr} \rangle$ INFIX1 $\langle \text{expr} \rangle$
		$\langle \text{expr} \rangle$ INFIX2 $\langle \text{expr} \rangle$
		$\langle \text{expr} \rangle$ INFIX3 $\langle \text{expr} \rangle$
		$\langle \text{expr} \rangle = \langle \text{expr} \rangle$
		$\langle \text{expr} \rangle :: \langle \text{expr} \rangle$
		$\langle \text{simple\_expr} \rangle$ [ $\langle \text{simple\_expr} \rangle$ ]
		<b>match</b> $\langle \text{expr} \rangle$ <b>with</b> $\langle \text{match\_case} \rangle_{\text{BAR}}^+$
$\langle \text{simple\_labeled\_expr} \rangle$	::=	IDENT : $\langle \text{simple\_expr} \rangle$
		$\langle \text{simple\_expr} \rangle$
		~ IDENT
$\langle \text{match\_case} \rangle$	::=	$\langle \text{pattern} \rangle \rightarrow \langle \text{expr} \rangle$
$\langle \text{simple\_expr} \rangle$	::=	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
		<b>true</b>
		<b>false</b>
$\langle \text{expr\_comma\_list} \rangle$	::=	$\langle \text{expr\_comma\_list} \rangle , \langle \text{expr} \rangle$
		$\langle \text{expr} \rangle , \langle \text{expr} \rangle$
$\langle \text{pattern} \rangle$	::=	$\langle \text{simple\_pattern} \rangle$
		$\langle \text{pattern\_comma\_list} \rangle$
		$\langle \text{pattern} \rangle :: \langle \text{pattern} \rangle$
		[ $\langle \text{simple\_pattern} \rangle^+ ;$ ]
$\langle \text{fun\_pattern} \rangle$	::=	IDENT
$\langle \text{simple\_pattern} \rangle$	::=	IDENT
		-
		( $\langle \text{pattern} \rangle$ )
		( )
		[ ]
$\langle \text{pattern\_comma\_list} \rangle$	::=	$\langle \text{pattern\_comma\_list} \rangle , \langle \text{pattern} \rangle$
		$\langle \text{pattern} \rangle , \langle \text{pattern} \rangle$
$\langle \text{struct\_graph\_desc} \rangle$	::=	$\langle \text{struct\_decl} \rangle^*$
$\langle \text{struct\_decl} \rangle$	::=	$\langle \text{wire\_decl} \rangle$
		$\langle \text{box\_decl} \rangle$
$\langle \text{wire\_decl} \rangle$	::=	<b>wire</b> IDENT* : $\langle \text{type\_expr} \rangle$
$\langle \text{box\_decl} \rangle$	::=	<b>box</b> IDENT : IDENT $\langle \text{box\_inps} \rangle$ $\langle \text{box\_outps} \rangle$

$$\begin{aligned}
\langle \text{box\_inps} \rangle &::= ( \langle \text{box\_inp} \rangle^* ) \\
\langle \text{box\_outps} \rangle &::= ( \langle \text{box\_outp} \rangle^* ) \\
\langle \text{box\_inp} \rangle &::= \text{IDENT} \\
&\quad | \quad ' \langle \text{basic\_expr} \rangle ' \\
\langle \text{box\_outp} \rangle &::= \text{IDENT} \\
\langle \text{basic\_expr} \rangle &::= \text{IDENT} \\
&\quad | \quad \langle \text{const\_expr} \rangle \\
&\quad | \quad \langle \text{basic\_expr} \rangle \text{ INFIX1 } \langle \text{basic\_expr} \rangle \\
&\quad | \quad \langle \text{basic\_expr} \rangle \text{ INFIX2 } \langle \text{basic\_expr} \rangle \\
&\quad | \quad \langle \text{basic\_expr} \rangle \text{ INFIX3 } \langle \text{basic\_expr} \rangle \\
&\quad | \quad ( \langle \text{basic\_expr} \rangle )
\end{aligned}$$

## Lexical Syntax

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