

# HoCL v1.1

## Syntax

J. Sérot



```

⟨program⟩ ::= ⟨top_decl⟩* EOF

⟨top_decl⟩ ::= ⟨type_decl⟩ ;
              | ⟨val_decl⟩ ;
              | ⟨node_decl⟩ ;

⟨type_decl⟩ ::= type IDENT

⟨node_decl⟩ ::= node IDENT [⟨node_param_decls⟩] in ⟨io_decls⟩ out ⟨io_decls⟩
              | [⟨node_impl⟩]
              | graph IDENT [⟨graph_param_decls⟩] in ⟨io_decls⟩ out ⟨io_decls⟩
              | ⟨node_impl⟩

⟨node_impl⟩ ::= fun ⟨val_decl⟩* end

⟨node_param_decls⟩ ::= param ( ⟨node_param_decl⟩*, )

⟨node_param_decl⟩ ::= IDENT : ⟨type_expr⟩

⟨graph_param_decls⟩ ::= param ( ⟨graph_param_decl⟩*, )

⟨graph_param_decl⟩ ::= IDENT : ⟨type_expr⟩ = ⟨simple_expr⟩

⟨io_decls⟩ ::= ( ⟨io_decl⟩*, )

⟨io_decl⟩ ::= IDENT : ⟨type_expr⟩

⟨type_expr⟩ ::= IDENT

⟨val_decl⟩ ::= val [rec] ⟨binding⟩and+

⟨binding⟩ ::= ⟨pattern⟩ = ⟨expr⟩
            | IDENT ⟨simple_pattern⟩+ = ⟨expr⟩

⟨expr⟩ ::= ⟨simple_expr⟩
          | ⟨simple_expr⟩ ⟨simple_expr⟩+
          | ⟨expr_comma_list⟩
          | fun ⟨pattern⟩ → ⟨expr⟩
          | let [rec] ⟨binding⟩and+ in ⟨expr⟩
          | if ⟨expr⟩ then ⟨expr⟩ else ⟨expr⟩
          | ⟨expr⟩ INFIX1 ⟨expr⟩
          | ⟨expr⟩ INFIX2 ⟨expr⟩
          | ⟨expr⟩ INFIX3 ⟨expr⟩
          | ⟨expr⟩ = ⟨expr⟩

⟨simple_expr⟩ ::= IDENT
              | ( ⟨expr⟩ )
              | ( )
              | ⟨const_expr⟩

⟨const_expr⟩ ::= INT
              | true
              | false

```

$$\begin{aligned}
\langle \text{expr\_comma\_list} \rangle &::= \langle \text{expr\_comma\_list} \rangle , \langle \text{expr} \rangle \\
&\quad | \quad \langle \text{expr} \rangle , \langle \text{expr} \rangle \\
\langle \text{pattern} \rangle &::= \langle \text{simple\_pattern} \rangle \\
&\quad | \quad \langle \text{pattern\_comma\_list} \rangle \\
\langle \text{simple\_pattern} \rangle &::= \text{IDENT} \\
&\quad | \quad - \\
&\quad | \quad ( \langle \text{pattern} \rangle ) \\
&\quad | \quad ( ) \\
\langle \text{pattern\_comma\_list} \rangle &::= \langle \text{pattern\_comma\_list} \rangle , \langle \text{pattern} \rangle \\
&\quad | \quad \langle \text{pattern} \rangle , \langle \text{pattern} \rangle
\end{aligned}$$

## Lexical Syntax

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