HoCL Manual - 1.0a

J. Sérot

Contents

Chapter 1

Syntax

This appendix gives a BNF definition of the concrete syntax for HoCL programs. The meta-syntax is conventional. Keywords are written in **bold** and non-terminals like ⟨this⟩. Vertical bars `|` are used to indicate alternatives. Constructs enclosed in brackets `[ ... ]` are optional. The notation ϵ denotes an empty construct. The notation $\mathrm{E}_{\mathrm{s}}^{*}$ (resp. $\mathrm{E}_{\mathrm{s}}^{+}$) denotes a list of zero (resp. one) or more elements E separated by s. Value-attributed terminals are denoted like *this*. The terminals *infix3*, *infix2* and *infix0* respectively correspond to infix operators {`*`,`/`,`%`}, {`+`,`-`} and {`@@`, `|>`, `|->`}. The other definitions (*ident*, *int*, *string*) are classical.

⟨program⟩ ::= ⟨decl⟩*

⟨decl⟩ ::= ⟨type_decl⟩ **;**
| ⟨value_decl⟩ **;**
| ⟨node_decl⟩ **;**
| ⟨graph_decl⟩ **;**

⟨type_decl⟩ ::= **type** *ident*

⟨value_decl⟩ ::= **val** [**rec**] ⟨net_binding⟩

⟨node_decl⟩ ::= ⟨node_intf⟩ ⟨node_impl⟩

⟨node_intf⟩ ::= **node** *ident* [⟨node_params⟩] **in** ⟨io_decls⟩ **out** ⟨io_decls⟩

⟨node_impl⟩ ::= ϵ
| **actor** ⟨actor_desc⟩* **end**
| **struct** ⟨struct_graph_desc⟩ **end**
| **fun** ⟨fun_graph_desc⟩ **end**

⟨actor_desc⟩ ::= *ident* **(** $\langle \text{impl\_attr} \rangle^{*}_{,}$ **)**

⟨impl_attr⟩ ::= *ident* **=** *string*
| *ident*

⟨node_params⟩ ::= **param (** $\langle \text{node\_param\_decl} \rangle^{*}_{,}$ **)**

⟨node_param_decl⟩ ::= *ident* **:** ⟨simple_type_expr⟩

⟨io_decls⟩ ::= **(** ⟨io_decl⟩$^*_,$ **)**

⟨io_decl⟩ ::= *ident* **:** ⟨simple_type_expr⟩ ⟨opt_io_annots⟩

⟨opt_io_annots⟩ ::= ϵ
| [⟨core_expr⟩]
| **{** ⟨io_annot⟩$^*_,$ **}**

⟨io_annot⟩ ::= **rate =** ⟨core_expr⟩
| **other =** *string*

⟨core_expr⟩ ::= ⟨simple_core_expr⟩
| ⟨core_expr⟩ *infix3* ⟨core_expr⟩
| ⟨core_expr⟩ *infix2* ⟨core_expr⟩
| ⟨core_expr⟩ $*$ ⟨core_expr⟩

⟨simple_core_expr⟩ ::= *ident*
| *int*
| **true**
| **false**
| **(** ⟨core_expr⟩ **)**

⟨simple_type_expr⟩ ::= *ident*
| **int**
| **bool**

⟨graph_decl⟩ ::= **graph** *ident* [⟨graph_params⟩] **in** ⟨io_decls⟩ **out** ⟨io_decls⟩ ⟨graph_defn⟩

⟨graph_params⟩ ::= **param (** ⟨graph_param_value⟩$^*_,$ **)**

⟨graph_param_value⟩ ::= *ident* **:** ⟨simple_type_expr⟩ **=** ⟨const_param_value⟩

⟨const_param_value⟩ ::= *int*
| **true**
| **false**

⟨graph_defn⟩ ::= **struct** ⟨struct_graph_desc⟩ **end**
| **fun** ⟨fun_graph_desc⟩ **end**

⟨struct_graph_desc⟩ ::= ⟨struct_defn⟩*

⟨struct_defn⟩ ::= ⟨gwire_defn⟩
| ⟨gnode_defn⟩

⟨gwire_defn⟩ ::= **wire** *ident*$^*_,$ **:** ⟨simple_type_expr⟩

⟨gnode_defn⟩ ::= **node** *ident* **:** *ident* [⟨gnode_params⟩] ⟨gnode_ios⟩ ⟨gnode_ios⟩

⟨gnode_params⟩ ::= < ⟨core_expr⟩$^{*}_{,}$ >

⟨gnode_ios⟩ ::= (⟨gnode_io⟩$^{*}_{,}$)

⟨gnode_io⟩ ::= *ident*

⟨fun_graph_desc⟩ ::= ⟨net_defn⟩*

⟨net_defn⟩ ::= **val** [**rec**] ⟨net_binding⟩$^{+}_{\textbf{and}}$

⟨net_binding⟩ ::= ⟨net_pattern⟩ = ⟨net_expr⟩
| ⟨net_binding_name⟩ ⟨simple_net_pattern⟩$^{+}$ = ⟨net_expr⟩

⟨net_binding_name⟩ ::= *ident*
| (*infix0*)

⟨net_expr⟩ ::= ⟨simple_net_expr⟩
| ⟨simple_net_expr⟩ ⟨simple_net_expr⟩$^{+}$
| ⟨net_expr_comma_list⟩
| ⟨net_expr⟩ **::** ⟨net_expr⟩
| ⟨simple_net_expr⟩ [⟨simple_net_expr⟩]
| **let** [**rec**] ⟨net_binding⟩$^{+}_{\textbf{and}}$ **in** ⟨net_expr⟩
| **fun** ⟨net_pattern⟩ → ⟨net_expr⟩
| **match** ⟨net_expr⟩ **with** ⟨net_case⟩$^{+}_{|}$
| **if** ⟨net_expr⟩ **then** ⟨net_expr⟩ **else** ⟨net_expr⟩
| ⟨net_expr⟩ *infix3* ⟨net_expr⟩
| ⟨net_expr⟩ *infix2* ⟨net_expr⟩
| ⟨net_expr⟩ *infix0* ⟨net_expr⟩
| ⟨net_expr⟩ > ⟨net_expr⟩
| ⟨net_expr⟩ < ⟨net_expr⟩
| ⟨net_expr⟩ ∗ ⟨net_expr⟩
| ⟨net_expr⟩ = ⟨net_expr⟩
| ⟨net_expr⟩ ≠ ⟨net_expr⟩

⟨simple_net_expr⟩ ::= *ident*
| *ident* < ⟨core_expr⟩$^{+}_{,}$ >
| ( )
| [⟨net_expr_comma_list⟩]
| [ ]
| *int*
| **true**
| **false**
| (⟨net_expr⟩)

⟨net_expr_comma_list⟩ ::= ⟨net_expr_comma_list⟩ , ⟨net_expr⟩
| ⟨net_expr⟩ , ⟨net_expr⟩

⟨net_case⟩ ::= ⟨net_pattern⟩ → ⟨net_expr⟩

⟨net_pattern⟩ ::= ⟨simple_net_pattern⟩
| ⟨net_pattern_comma_list⟩
| ⟨net_pattern⟩ **::** ⟨net_pattern⟩
| [⟨net_pattern_comma_list⟩]

⟨simple_net_pattern⟩ ::= *ident*
| _
| **(** ⟨net_pattern⟩ **)**
| [ ]
| **( )**

⟨net_pattern_comma_list⟩ ::= ⟨net_pattern_comma_list⟩ **,** ⟨net_pattern⟩
| ⟨net_pattern⟩ **,** ⟨net_pattern⟩