# The Cedilleum Language Specification

Syntax, Typing, Reduction, and Elaboration

Christopher Jenkins

June 15, 2018

## 1 Syntax

| | | | |
|---|---|---|---|
| $id$ | | | identifiers for definitions |
| $u$ | | | term variables |
| $X$ | | | type variables |
| $k$ | | | kind variables |
| $x$ | $::=$ | $id \mid u \mid X \mid k$ | any variable |

Figure 1: Identifiers

$$
\begin{array}{rcl}
uterms & ::= & u \\
 & & \lambda\, u.\, uterm \\
 & & uterm\ uterm
\end{array}
$$

Figure 2: Untyped terms

| | | | |
|---|---|---|---|
| $mod$ | ::= | **module** $id$ **.** $imprt^*$ $cmd^*$ | module declarations |
| $imprt$ | ::= | **import** $id$ **.** | module imports |
| $cmd$ | ::= | $defTermOrType$ | definitions |
| | | $defDataType$ | |
| | | $defKind$ | |
| $defTermOrType$ | ::= | $id$ $checkType^?$ $=$ $term$ **.** | term definition |
| | | $id \triangleleft kind = type$ **.** | type definition |
| $defDataType$ | ::= | **data** $id$ $param^*$ **:** $kind = constr^*$ **.** | datatype definitions |
| $defKind$ | ::= | $k$ $params^* = kind$ | |
| $checkType$ | ::= | $\triangleleft$ $type$ | annotation for term definition |
| $param$ | ::= | $(x$ **:** $typeOrKind)$ | |
| $typeOrKind$ | ::= | $type$ | |
| | | $kind$ | |
| $constr$ | ::= | $\mid$ $id$ **:** $type$ | |

Figure 3: Modules and definitions

| | | | |
|---|---|---|---|
| $kind$ | ::= | $\star$ | |
| | | $\Pi\ x$ **:** $typeOrKind$ **.** $kind$ | explicit product |
| | | $typeOrKind \to kind$ | kind arrow |
| | | $k$ $term$ | |
| | | $k \cdot type$ | |
| $type$ | ::= | $X$ | |
| | | $\Pi\ x$ **:** $typeOrKind$ **.** $type$ | explicit product |
| | | $\forall\ x$ **:** $typeOrKind$ **.** $type$ | implicit product |
| | | $\lambda\ x$ **:** $typeOrKind$ **.** $type$ | type-level function |
| | | $type \to type$ | normal arrow type |
| | | $type \Rightarrow type$ | arrow with erased domain |
| | | $\{\ uterm\ \simeq\ uterm\}$ | untyped equality |

Figure 4: Kinds and types

| | | | |
|---|---|---|---|
| *term* | ::= | $x$ | |
| | | $\lambda\ x\ class^?\ \mathbf{.}\ term$ | normal abstraction |
| | | $\Lambda\ x\ class^?\ \mathbf{.}\ term$ | erased abstraction |
| | | $[\ defTermOrType\ ]\ \mathbf{-}\ term$ | let |
| | | $term\ arg^*$ | applications |
| | | $\beta\mathbf{<}term\mathbf{>}\ \mathbf{\{}term\mathbf{\}}$ | reflexivity of equality |
| | | $\varsigma\ term$ | symmetry of equality |
| | | $\rho\ term\ guide^?\ \mathbf{-}\ term$ | equality elimination by type-guided rewriting |
| | | $\delta\ type^?\ \mathbf{-}\ term$ | ex falso quodlibet |
| | | $\phi\ term\ \mathbf{-}\ term\ \mathbf{\{}term\mathbf{\}}$ | cast |
| | | $\mu\ term\ motive^?\ \mathbf{\{}\ case^*\ \mathbf{\}}$ | pattern match and fixpoint |
| *arg* | ::= | *term* | normal application |
| | | $\mathbf{-}\ term$ | application to erased term |
| | | $\cdot\ term$ | application to type |
| *class* | ::= | $\mathbf{:}\ typeOrKind$ | |
| *guide* | ::= | $\mathbf{@}\ x\mathbf{.}\ type$ | guide for equality rewrite |
| *motive* | ::= | $\mathbf{@}\ type$ | motive for induction |
| *case* | ::= | $\mid\ id\ arg^*\ \mathbf{.}\ term$ | |

Figure 5: Annotated Terms