The Cedilleum Language Specification
Syntax, Typing, Reduction, and Elaboration

Christopher Jenkins

June 13, 2018

1   Syntax

$$
\begin{array}{lll}
id & & \text{identifiers for definitions} \\
u & & \text{term variables} \\
X & & \text{type variables} \\
k & & \text{kind variables} \\
x & ::= \quad id \mid u \mid X \mid k & \text{any variable}
\end{array}
$$

(a) Identifiers

$$
\begin{array}{llll}
mod & ::= & \textbf{module } id \textbf{ . } imprt^* \; cmd^* & \text{module declarations} \\
imprt & ::= & \textbf{import } id \textbf{ .} & \text{module imports} \\
cmd & ::= & defTermOrType \mid defDataType \mid defKind & \text{definitions} \\
\\
defTermOrType & ::= & id \; chkType^? = term \textbf{ .} \mid id \vartriangleleft kind = type \textbf{ .} & \text{term and type definitions} \\
checkType & ::= & \vartriangleleft type & \text{annotation for term definition} \\
defDataType & ::= & \textbf{data } id \; param^* \textbf{ : } kind = constr^* \textbf{ .} & \text{datatype definitions} \\
param & ::= & (x : typeOrKind) & \\
constr & ::= & \mid id : type & \\
defKind & ::= & k \; params^* = kind & \\
\\
arg & ::= & term \mid \cdot \; type & \\
kind & ::= & \star & \\
& & \Pi \; x : typeOrKind \textbf{ .} \; kind & \\
& & kind \rightarrow kind & \\
& & type \rightarrow kind & \\
& & k \; arg^* &
\end{array}
$$

Figure 1: Syntax for Cedilleum Programs