

1 Grammar

\mathbb{S}	<i>System</i>
\mathbb{S}	<i>Subsystem Name</i>
\mathbb{E}	<i>Explanation</i>
\mathbb{C}	<i>Component Name</i>
\mathbf{q}	<i>Query</i>
\mathbf{c}	<i>Command</i>
\mathbf{i}	<i>Constraint</i>
<hr/>	
$\mathbb{S} \leftarrow \mathbb{E}$	<i>Subsystem Explanation</i>
$\mathbb{S} \leftarrow \mathbb{S}$	<i>Subsystem Containment</i>
$\mathbb{C} \leftarrow \mathbb{E}$	<i>Intro Explanation</i>
$\mathbb{C} \leftarrow \mathbb{S}$	<i>Intro Containment</i>

2 Type System

2.1 Type Grammar

$\tau ::= \hat{\tau} \mid \check{\tau}$	<i>Type</i>
$\hat{\tau} ::= \mathcal{B} \mid \mathcal{Z} \mid \mathcal{N} \mid \textit{System} \mid \textit{Subsystem} \mid \textit{Component}$	<i>Primitive Type (TODO: More!)</i>
$\check{\tau} ::= \check{\tau}_1, \check{\tau}_2, \dots$	<i>Derived Type</i>

2.2 Type Environment

In order to define the type system, we must first introduce a few auxilliary structures:

$\mathbb{CI} ::= \{ \mathbb{C} \mapsto \mathbb{C} \}$	<i>Component Inheritance Map</i>
$\mathbb{CT} ::= \{ \mathbb{C} \mapsto \check{\tau} \}$	<i>Component Type Map</i>
$\mathbb{CD} ::= \{ \check{\tau} \mapsto \langle \mathbb{S}, \mathbb{E}, \{ \mathbf{f} \} \rangle \}$	<i>Component Data Map</i>
$\mathbf{f} ::= \mathbf{q} \mid \mathbf{c} \mid \mathbf{i}$	<i>Feature</i>
<hr/>	
$\mathbb{ST} ::= \{ \mathbb{S} \mapsto \check{\tau} \}$	<i>Subsystem Type Map</i>
$\mathbb{SD} ::= \{ \check{\tau} \mapsto \langle \mathbb{SP}, \mathbb{E}, \{ \mathbb{S} \}, \{ \mathbb{C} \} \rangle \}$	<i>Subsystem Data Map</i>
$\mathbb{SP} ::= \mathbb{S} \mid \mathbb{S}$	<i>Parent of Subsystem</i>
<hr/>	
$\Gamma ::= \langle \mathbb{CI}, \mathbb{CT}, \mathbb{CD}, \mathbb{T}, \mathbb{ST}, \mathbb{SD} \rangle$	<i>Type Environment</i>

We will need more than this in the type environment. This is to get started!

The type environment consists of a number of distinct elements required by our type system definition. For simplifying the rules, we define a short-hand notation for each of them.

$$\begin{aligned}
\mathbb{CI}^\Gamma &\doteq \mathbb{CI} && \text{where } \Gamma = \langle \mathbb{CI}, \mathbb{CT}, \mathbb{CD}, \mathbb{T}, \mathbb{ST}, \mathbb{SD} \rangle \\
\mathbb{CT}^\Gamma &\doteq \mathbb{CT} && \text{where } \Gamma = \langle \mathbb{CI}, \mathbb{CT}, \mathbb{CD}, \mathbb{T}, \mathbb{ST}, \mathbb{SD} \rangle \\
\mathbb{CD}^\Gamma &\doteq \mathbb{CD} && \text{where } \Gamma = \langle \mathbb{CI}, \mathbb{CT}, \mathbb{CD}, \mathbb{T}, \mathbb{ST}, \mathbb{SD} \rangle \\
\mathbb{T}^\Gamma &\doteq \mathbb{T} && \text{where } \Gamma = \langle \mathbb{CI}, \mathbb{CT}, \mathbb{CD}, \mathbb{T}, \mathbb{ST}, \mathbb{SD} \rangle \\
\mathbb{ST}^\Gamma &\doteq \mathbb{ST} && \text{where } \Gamma = \langle \mathbb{CI}, \mathbb{CT}, \mathbb{CD}, \mathbb{T}, \mathbb{ST}, \mathbb{SD} \rangle \\
\mathbb{SD}^\Gamma &\doteq \mathbb{SD} && \text{where } \Gamma = \langle \mathbb{CI}, \mathbb{CT}, \mathbb{CD}, \mathbb{T}, \mathbb{ST}, \mathbb{SD} \rangle
\end{aligned}$$

2.3 Rules

Currently all rules have the form: $e : \check{\tau}$. This is not strictly necessary; for example many rules can be relations that does not necessarily involve types at all (e.g. Intro explanation). All we have to ensure is that the set of rules are inductive in the right way. But since I haven't worked out all the rules, I am keeping the rules consistent and always including a type for the time being. That can be fixed later.

$$\begin{array}{c}
\text{SYSTEM INTRODUCTION} \\
\hline
\mathbb{S} : \text{System} \\
\\
\text{SUBSYSTEM INTRODUCTION} \\
\hline
\check{\tau} <: \text{Subsystem} \\
\hline
\mathbb{S} : \check{\tau} \\
\\
\text{SUBSYSTEM EXPLANATION INTRODUCTION} \\
\hline
\mathbb{S} \mapsto \check{\tau} \in \mathbb{ST}^\Gamma \quad \check{\tau} \mapsto \langle \mathbb{SP}, \mathbb{E}, \{\mathbb{S}\}, \{\mathbb{C}\} \rangle \in \mathbb{SD}^\Gamma \quad \mathbb{E} = \mathbb{E}' \\
\hline
\mathbb{S} \leftarrow \mathbb{E}' : \check{\tau} \\
\\
\text{SUBSYSTEM CONTAINMENT} \\
\hline
\mathbb{S} \mapsto \check{\tau} \in \mathbb{ST}^\Gamma \quad \check{\tau} \mapsto \langle \mathbb{SP}, \mathbb{E}, \{\mathbb{S}\}, \{\mathbb{C}\} \rangle \in \mathbb{SD}^\Gamma \quad \mathbb{SP} = \mathbb{S}' \\
\hline
\mathbb{S} \leftarrow \mathbb{S}' : \check{\tau} \\
\\
\text{COMPONENT INTRODUCTION} \\
\hline
\mathbb{C} \mapsto \check{\tau} \in \mathbb{CT}^\Gamma \\
\hline
\Gamma \vdash \mathbb{C} : \check{\tau} \\
\\
\text{COMPONENT SUBTYPING} \\
\hline
\mathbb{C} \mapsto \check{\tau} \in \mathbb{CT}^\Gamma \quad \mathbb{C} \mapsto \mathbb{C}' \in \mathbb{CI}^\Gamma \\
\mathbb{C}' \mapsto \check{\tau}' \in \mathbb{CT}^\Gamma \quad \text{NCI}(\mathbb{CI}^\Gamma, \mathbb{C}, \mathbb{C}') \quad \check{\tau} <: \check{\tau}' \in \mathbb{T}^\Gamma \\
\hline
\Gamma \vdash \mathbb{C} : \check{\tau} \\
\\
\text{COMPONENT IMPLICIT SUBTYPING} \\
\hline
\mathbb{C} \mapsto \check{\tau} \in \mathbb{CT}^\Gamma \quad \nexists \mathbb{C}'. \mathbb{C} \mapsto \mathbb{C}' \in \mathbb{CI}^\Gamma \quad \check{\tau} <: \text{Component} \in \mathbb{T}^\Gamma \\
\hline
\Gamma \vdash \mathbb{C} : \check{\tau}
\end{array}$$

$$\begin{array}{c}
\text{COMPONENT EXPLANATION INTRODUCTION} \\
\hline
\frac{C \mapsto \check{\tau} \in \mathbb{CT}^\Gamma \quad \check{\tau} \mapsto \langle S, E, \{f\} \rangle \in \mathbb{CD}^\Gamma \quad E = E'}{C \leftarrow E' : \check{\tau}} \\
\\
\text{COMPONENT CONTAINMENT} \\
\hline
\frac{\begin{array}{l} C \mapsto \check{\tau} \in \mathbb{CT}^\Gamma \quad \check{\tau} \mapsto \langle S, E, \{f\} \rangle \in \mathbb{CD}^\Gamma \quad S = S' \\ S' \mapsto \check{\tau}' \in \mathbb{ST}^\Gamma \quad \check{\tau}' \mapsto \langle SP, E, \{S\}, \{C\} \rangle \in \mathbb{SD}^\Gamma \quad C \in \{C\} \end{array}}{C \leftarrow S' : \check{\tau}}
\end{array}$$

2.4 Auxiliary Functions and Relations

Non-circular-Inheritance indicates that a component's inheritance chain does not have a cycle. Note that the particular relation only checks for circles that includes the specific component. However since this relation must be satisfied by every component, any cycle in the chain will be detected at some point. Formally non-circularity is a ternary relation defined inductively as below:

$$\text{NCI}(\mathbb{CI}^\Gamma, C, C') = \begin{cases} \text{undefined iff } C = C' \\ \text{undefined iff } \text{NCI}(\mathbb{CI}^\Gamma, C, C'') \text{ where } C' \mapsto C'' \in \mathbb{CI}^\Gamma \\ \text{defined otherwise} \end{cases}$$