

The : and :: Relations

$$\frac{\text{triv } x}{x : \text{Triv}} \quad (1)$$

Fonts <hr style="width: 100%;"/> \mathbf{X}, \mathbf{x} - Concrete syntax \mathbf{X}, \mathbf{x} - Syntax variables \mathcal{X}, χ - Type variables <hr style="width: 100%;"/>	$\frac{\text{triv } x}{x :: \text{Box}(\text{Triv})} \quad (2)$
---	---

$$\frac{\text{int } x}{x : \text{Int}} \quad (3)$$

$$\frac{\text{int } x}{x :: \text{Box}(\text{Int})} \quad (4)$$

$$\frac{x :: \text{Box}(\mathcal{T})}{\text{immut } x :: \text{IBox}(\mathcal{T})} \quad (5)$$

$$\frac{x :: \mathcal{T}}{\text{ref } x :: \text{Box}(\text{Ref}(\mathcal{T}))} \quad (6)$$

$$\frac{x :: \mathcal{T}}{\& x : \text{Ref}(\mathcal{T})} \quad (7)$$

$$\frac{x : \text{Ref}(\mathcal{T})}{x @ : \mathcal{T}} \quad (8)$$

$$\frac{x : \text{Ref}(\mathcal{T})}{x @ :: \mathcal{T}} \quad (9)$$

$$\frac{x :: \mathcal{T} \quad y :: \mathcal{U} \quad \mathcal{T} <:: \mathcal{U} \quad x : \mathcal{V}}{x := y : \mathcal{V}} \quad (10)$$

$x : \mathcal{T}$ is read as “expression x is of type \mathcal{T} and is in an *r-context*.”

$x :: \mathcal{T}$ is read as “variable x is of type \mathcal{T} and is in an *l-context*.”

The operators could also be referred to as the “r-type of” and “l-type of” operators.

l-context denotes everything that is *assignable* (indicated as a storable memory). r-context, on the other hand, denotes everything that is *expressible* (can be produced by an expression).

There is no r-value (e.g. expression) of the type $\text{Box}(\mathcal{T})$.

We omit rules for *Con* types as they only operate on r-values.

We omit rules for *Fun* types as they only accept r-values. Any variable and/or primitive type has both r-value and l-value (when it comes to primitive types, only r-value). In all cases, the r-value part of the actual parameter is passed when the function is being called.

The $<::$ Relation

$$\frac{}{\mathcal{Box}(Triv) <:: \mathcal{Box}(Triv)} \quad (11)$$

$$\frac{}{\mathcal{Box}(Int) <:: \mathcal{Box}(Int)} \quad (12)$$

$$\frac{\mathcal{T} <:: \mathcal{Box}(\mathcal{U})}{\mathcal{T} <:: I\mathcal{Box}(\mathcal{U})} \quad (13)$$

$$\frac{\mathcal{Box}(\mathcal{T}) <:: I\mathcal{Box}(\mathcal{U})}{I\mathcal{Box}(\mathcal{T}) <:: I\mathcal{Box}(\mathcal{U})} \quad (14)$$

$$\frac{\mathcal{T} <:: \mathcal{U}}{\mathcal{Ref} \ \mathcal{T} <:: \mathcal{Ref} \ \mathcal{U}} \quad (15)$$

$<::$ specifies the relationship between two container types.

$\mathcal{Box}?$ statement checks for both mutable and immutable containers. In other words, $\mathcal{Box}(Int)$ and $I\mathcal{Box}(Int)$ would both satisfy the $\mathcal{Box}?$ condition.