

Gradual Intersection Types

Pedro Ângelo, Mário Florido

March 8, 2018

1 Language Definition

Syntax

Types $T ::= Int \mid Bool \mid T \rightarrow T \mid T \cap \dots \cap T$
Expressions $e ::= x \mid \lambda x : T . e \mid e e \mid n \mid true \mid false$

$\boxed{\Gamma \vdash_{\cap S} e : T}$ Typing

$$\begin{array}{c} \frac{x : T \in \Gamma}{\Gamma \vdash_{\cap S} x : T} \text{ T-VAR} \qquad \frac{\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap S} e : T}{\Gamma \vdash_{\cap S} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T} \text{ T-ABS} \\[10pt] \frac{\Gamma, x : T_i \vdash_{\cap S} e : T}{\Gamma \vdash_{\cap S} \lambda x : T_1 \cap \dots \cap T_n . e : T_i \rightarrow T} \text{ T-ABS'} \\[10pt] \frac{\Gamma \vdash_{\cap S} e_1 : T_1 \cap \dots \cap T_n \rightarrow T \quad \Gamma \vdash_{\cap S} e_2 : T_1 \cap \dots \cap T_n}{\Gamma \vdash_{\cap S} e_1 e_2 : T} \text{ T-APP} \\[10pt] \frac{\Gamma \vdash_{\cap S} e : T_1 \quad \dots \quad \Gamma \vdash_{\cap S} e : T_n}{\Gamma \vdash_{\cap S} e : T_1 \cap \dots \cap T_n} \text{ T-GEN} \quad \frac{\Gamma \vdash_{\cap S} e : T_1 \cap \dots \cap T_n}{\Gamma \vdash_{\cap S} e : T_i} \text{ T-INST} \quad \frac{}{\Gamma \vdash_{\cap S} n : Int} \text{ T-INT} \\[10pt] \frac{}{\Gamma \vdash_{\cap S} true : Bool} \text{ T-TRUE} \qquad \frac{}{\Gamma \vdash_{\cap S} false : Bool} \text{ T-FALSE} \end{array}$$

Figure 1: Static Intersection Type System ($\vdash_{\cap S}$)

Syntax

Types $T ::= Int \mid Bool \mid Dyn \mid T \rightarrow T \mid T \cap \dots \cap T$
Expressions $e ::= x \mid \lambda x : T . e \mid e e \mid n \mid true \mid false$

$\boxed{\Gamma \vdash_{\cap G} e : T}$ Typing

$$\begin{array}{c}
\frac{x : T \in \Gamma}{\Gamma \vdash_{\cap G} x : T} \text{ T-VAR} \qquad \frac{\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap G} e : T}{\Gamma \vdash_{\cap G} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T} \text{ T-ABS} \\
\\
\frac{\Gamma, x : T_i \vdash_{\cap G} e : T}{\Gamma \vdash_{\cap G} \lambda x : T_1 \cap \dots \cap T_n . e : T_i \rightarrow T} \text{ T-ABS'} \\
\\
\frac{\Gamma \vdash_{\cap G} e_1 : PM \quad PM \triangleright T_1 \cap \dots \cap T_n \rightarrow T \quad \Gamma \vdash_{\cap G} e_2 : T'_1 \cap \dots \cap T'_n \quad T'_1 \cap \dots \cap T'_n \sim T_1 \cap \dots \cap T_n}{\Gamma \vdash_{\cap G} e_1 e_2 : T} \text{ T-APP} \\
\\
\frac{\Gamma \vdash_{\cap G} e : T_1 \dots \Gamma \vdash_{\cap G} e : T_n}{\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n} \text{ T-GEN} \qquad \frac{\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n}{\Gamma \vdash_{\cap G} e : T_i} \text{ T-INST} \\
\\
\frac{}{\Gamma \vdash_{\cap G} n : Int} \text{ T-INT} \qquad \frac{}{\Gamma \vdash_{\cap G} true : Bool} \text{ T-TRUE} \qquad \frac{}{\Gamma \vdash_{\cap G} false : Bool} \text{ T-FALSE}
\end{array}$$

$\boxed{T \sim T}$ Consistency

$$T \sim T \quad T \sim Dyn \quad Dyn \sim T \quad \frac{T_1 \sim T_3 \quad T_2 \sim T_4}{T_1 \rightarrow T_2 \sim T_3 \rightarrow T_4} \quad \frac{T_1 \sim T'_1 \dots T_n \sim T'_n}{T_1 \cap \dots \cap T_n \sim T'_1 \cap \dots \cap T'_n}$$

$\boxed{T \triangleright T}$ Pattern Matching

$$T_1 \rightarrow T_2 \triangleright T_1 \rightarrow T_2 \qquad Dyn \triangleright Dyn \rightarrow Dyn$$

Figure 2: Gradual Intersection Type System ($\vdash_{\cap G}$)

$\boxed{T \sqsubseteq T}$ Type Precision

$$\begin{array}{c}
\text{Dyn} \sqsubseteq T \quad T \sqsubseteq T \quad \frac{T_1 \sqsubseteq T_3 \quad T_2 \sqsubseteq T_4}{T_1 \rightarrow T_2 \sqsubseteq T_3 \rightarrow T_4} \quad \frac{T_1 \sqsubseteq T'_1 \dots T_n \sqsubseteq T'_n}{T_1 \cap \dots \cap T_n \sqsubseteq T'_1 \cap \dots \cap T'_n} \\
\\
\frac{T \sqsubseteq T_1 \dots T \sqsubseteq T_n}{T \sqsubseteq T_1 \cap \dots \cap T_n} \quad \frac{T_1 \sqsubseteq T \dots T_n \sqsubseteq T}{T_1 \cap \dots \cap T_n \sqsubseteq T}
\end{array}$$

$\boxed{c \sqsubseteq c}$ Cast Precision

$$\begin{array}{c}
\frac{c \sqsubseteq c' \quad T_1 \sqsubseteq T'_1 \quad T_2 \sqsubseteq T'_2}{c : T_1 \Rightarrow^l T_2 \text{ }^{cl} \sqsubseteq c' : T'_1 \Rightarrow^{l'} T'_2 \text{ }^{cl'}} \quad \frac{c \sqsubseteq c' \quad \vdash_{\cap IC} c' : T \quad T_1 \sqsubseteq T \quad T_2 \sqsubseteq T}{c : T_1 \Rightarrow^l T_2 \text{ }^{cl} \sqsubseteq c'} \\
\\
\frac{c \sqsubseteq c' \quad \vdash_{\cap IC} c : T \quad T \sqsubseteq T_1 \quad T \sqsubseteq T_2}{c \sqsubseteq c' : T_1 \Rightarrow^l T_2 \text{ }^{cl}} \quad \frac{T_I \sqsubseteq T'_I \quad T_F \sqsubseteq T'_F}{\text{blame } T_I \text{ } T_F \text{ } l \text{ }^{cl} \sqsubseteq \text{blame } T'_I \text{ } T'_F \text{ } l' \text{ }^{cl'}} \\
\\
\frac{T \sqsubseteq T'}{\emptyset \text{ } T \text{ }^{cl} \sqsubseteq \emptyset \text{ } T' \text{ }^{cl'}}
\end{array}$$

$\boxed{e \sqsubseteq e}$ Expression Precision

$$\begin{array}{c}
x \sqsubseteq x \quad \frac{T \sqsubseteq T' \quad e \sqsubseteq e'}{\lambda x : T . e \sqsubseteq \lambda x : T' . e'} \quad \frac{e_1 \sqsubseteq e'_1 \quad e_2 \sqsubseteq e'_2}{e_1 \text{ } e_2 \sqsubseteq e'_1 \text{ } e'_2} \quad n \sqsubseteq n \quad \text{true} \sqsubseteq \text{true} \\
\\
\text{false} \sqsubseteq \text{false} \quad \frac{e \sqsubseteq e' \quad c_1 \sqsubseteq c'_1 \dots c_n \sqsubseteq c'_n}{e : c_1 \cap \dots \cap c_n \sqsubseteq e' : c'_1 \cap \dots \cap c'_n} \\
\\
\frac{e \sqsubseteq e' \quad \Gamma \vdash_{\cap CC} e' : T \quad \vdash_{\cap IC} c_1 : T_1 \dots \vdash_{\cap IC} c_n : T_n \quad T_1 \cap \dots \cap T_n \sqsubseteq T}{e : c_1 \cap \dots \cap c_n \sqsubseteq e'} \\
\\
\frac{e \sqsubseteq e' \quad \Gamma \vdash_{\cap CC} e : T \quad \vdash_{\cap IC} c_1 : T_1 \dots \vdash_{\cap IC} c_n : T_n \quad T \sqsubseteq T_1 \cap \dots \cap T_n}{e \sqsubseteq e' : c_1 \cap \dots \cap c_n} \\
\\
\frac{\Gamma \vdash_{\cap CC} e : T \quad T \sqsubseteq T'}{e \sqsubseteq \text{blame}_{T'} l}
\end{array}$$

Figure 3: Precision (\sqsubseteq)

Syntax

Types $T ::= Int \mid Bool \mid Dyn \mid T \rightarrow T$
Casts $c ::= c : T \Rightarrow^l T^{cl} \mid blame\ T\ T\ l^{cl} \mid \emptyset\ T^{cl}$

$\boxed{\vdash_{\cap IC} c : T}$ Typing

$$\frac{\vdash_{\cap IC} c : T_1 \quad T_1 \sim T_2}{\vdash_{\cap IC} (c : T_1 \Rightarrow^l T_2^{cl}) : T_2} \text{T-SINGLEIC} \quad \frac{}{\vdash_{\cap IC} blame\ T_I\ T_F\ l^{cl} : T_F} \text{T-BLAMEIC}$$

$$\frac{}{\vdash_{\cap IC} \emptyset\ T^{cl} : T} \text{T-EMPTYIC}$$

Figure 4: Intersection Casts Type System ($\vdash_{\cap IC}$)

Syntax

Types $T ::= Int \mid Bool \mid Dyn \mid T \rightarrow T \mid T \cap \dots \cap T$
Expressions $e ::= x \mid \lambda x : T . e \mid e\ e \mid n \mid true \mid false \mid e : c \cap \dots \cap c \mid blame_T\ l$

$\boxed{\Gamma \vdash_{\cap CC} e : T}$ Typing

Static Intersection Type System ($\vdash_{\cap S}$) *rules and*

$$\frac{\Gamma \vdash_{\cap CC} e_1 : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2} \quad \Gamma \vdash_{\cap CC} e_2 : T_{11} \cap \dots \cap T_{n1}}{\Gamma \vdash_{\cap CC} e_1\ e_2 : T_{12} \cap \dots \cap T_{n2}} \text{T-APP},$$

$$\frac{\Gamma \vdash_{\cap CC} e : T'_1 \cap \dots \cap T'_n \quad \vdash_{\cap IC} c_1 : T_1 \dots \vdash_{\cap IC} c_n : T_n \quad T'_1 \cap \dots \cap T'_n = initialType(c_1) \cap \dots \cap initialType(c_n)}{\Gamma \vdash_{\cap CC} e : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n} \text{T-INTERSECTIONCAST}$$

$$\frac{}{\Gamma \vdash_{\cap CC} blame_T\ l : T} \text{T-BLAME}$$

$\boxed{initialType(c) = T}$

$\boxed{finalType(c) = T}$

$initialType(c : T_1 \Rightarrow^l T_2^{cl}) = initialType(c)$

$finalType(c : T_1 \Rightarrow^l T_2^{cl}) = T_2$

$initialType(\emptyset\ T^{cl}) = T$

$finalType(\emptyset\ T^{cl}) = T$

$initialType(blame\ T_I\ T_F\ l^{cl}) = T_I$

$finalType(blame\ T_I\ T_F\ l^{cl}) = T_F$

Figure 5: Intersection Cast Calculus ($\vdash_{\cap CC}$)

$\boxed{\Gamma \vdash_{\text{NCC}} e \rightsquigarrow e : T}$ Compilation

$$\begin{array}{c}
\frac{x : T \in \Gamma}{\Gamma \vdash_{\text{NCC}} x \rightsquigarrow x : T} \text{C-VAR} \\
\\
\frac{\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\text{NCC}} e \rightsquigarrow e' : T}{\Gamma \vdash_{\text{NCC}} (\lambda x : T_1 \cap \dots \cap T_n . e) \rightsquigarrow (\lambda x : T_1 \cap \dots \cap T_n . e') : T_1 \cap \dots \cap T_n \rightarrow T} \text{C-ABS} \\
\\
\frac{\Gamma, x : T_i \vdash_{\text{NCC}} e \rightsquigarrow e' : T}{\Gamma \vdash_{\text{NCC}} (\lambda x : T_1 \cap \dots \cap T_n . e) \rightsquigarrow (\lambda x : T_1 \cap \dots \cap T_n . e') : T_i \rightarrow T} \text{C-ABS}' \\
\\
\frac{\Gamma \vdash_{\text{NCC}} e_1 \rightsquigarrow e'_1 : PM \quad PM \triangleright T_1 \cap \dots \cap T_n \rightarrow T \quad \Gamma \vdash_{\text{NCC}} e_2 \rightsquigarrow e'_2 : T'_1 \cap \dots \cap T'_n}{\begin{array}{c} T'_1 \cap \dots \cap T'_n \sim T_1 \cap \dots \cap T_n \quad PM \trianglelefteq S_1 \quad T_1 \cap \dots \cap T_n \rightarrow T \trianglelefteq S_2 \\ T'_1 \cap \dots \cap T'_n \trianglelefteq S_3 \quad T_1 \cap \dots \cap T_n \trianglelefteq S_4 \quad S_1, S_2, e'_1 \hookrightarrow e''_1 \quad S_3, S_4, e'_2 \hookrightarrow e''_2 \end{array}} \text{C-APP} \\
\Gamma \vdash_{\text{NCC}} e_1 e_2 \rightsquigarrow e''_1 e''_2 : T \\
\\
\frac{\Gamma \vdash_{\text{NCC}} e \rightsquigarrow e' : T_1 \dots \Gamma \vdash_{\text{NCC}} e \rightsquigarrow e' : T_n}{\Gamma \vdash_{\text{NCC}} e \rightsquigarrow e' : T_1 \cap \dots \cap T_n} \text{C-GEN} \quad \frac{\Gamma \vdash_{\text{NCC}} e \rightsquigarrow e' : T_1 \cap \dots \cap T_n}{\Gamma \vdash_{\text{NCC}} e \rightsquigarrow e' : T_i} \text{C-INST} \\
\\
\frac{}{\Gamma \vdash_{\text{NCC}} n \rightsquigarrow n : \text{Int}} \text{C-INT} \quad \frac{}{\Gamma \vdash_{\text{NCC}} \text{true} \rightsquigarrow \text{true} : \text{Bool}} \text{C-TRUE} \\
\\
\frac{}{\Gamma \vdash_{\text{NCC}} \text{false} \rightsquigarrow \text{false} : \text{Bool}} \text{C-FALSE}
\end{array}$$

$\boxed{T \trianglelefteq \{T\}}$ Instances

$$\begin{array}{ccc}
\text{Int} \trianglelefteq \{\text{Int}\} & \text{Bool} \trianglelefteq \{\text{Bool}\} & \text{Dyn} \trianglelefteq \{\text{Dyn}\} \\
\\
\frac{T_1 \trianglelefteq \{T_{11}, \dots, T_{1n}\}}{T_1 \rightarrow T_2 \trianglelefteq \{T_{11} \rightarrow T_2, \dots, T_{1n} \rightarrow T_2\}} & \frac{T_1 \trianglelefteq \{T_{11}, \dots, T_{1m}\} \dots T_n \trianglelefteq \{T_{n1}, \dots, T_{nj}\}}{T_1 \cap \dots \cap T_n \trianglelefteq \{T_{11}, \dots, T_{1m}, \dots, T_{n1}, \dots, T_{nj}\}}
\end{array}$$

$\boxed{S, S, e \hookrightarrow e}$

$$\begin{array}{c}
\{T_1\}, \{T_2\}, e \hookrightarrow e : (\emptyset T_1^0 : T_1 \Rightarrow^l T_2^0) \\
\\
\{T_{11}, \dots, T_{1n}\}, \{T_{21}, \dots, T_{2n}\}, e \hookrightarrow e : (\emptyset T_{11}^0 : T_{11} \Rightarrow^{l_1} T_{21}^0) \cap \dots \cap (\emptyset T_{1n}^0 : T_{1n} \Rightarrow^{l_n} T_{2n}^0) \\
\\
\{T_{11}, \dots, T_{1n}\}, \{T_2\}, e \hookrightarrow e : (\emptyset T_{11}^0 : T_{11} \Rightarrow^{l_1} T_2^0) \cap \dots \cap (\emptyset T_{1n}^0 : T_{1n} \Rightarrow^{l_n} T_2^0) \\
\\
\{T_1\}, \{T_{21}, \dots, T_{2n}\}, e \hookrightarrow e : (\emptyset T_1^0 : T_1 \Rightarrow^{l_1} T_{21}^0) \cap \dots \cap (\emptyset T_1^0 : T_1 \Rightarrow^{l_n} T_{2n}^0)
\end{array}$$

Figure 6: Compilation to the Cast Calculus

Syntax

Types $T ::= Int \mid Bool \mid Dyn \mid T \rightarrow T$
Ground Types $G ::= Int \mid Bool \mid Dyn \rightarrow Dyn$
Casts $c ::= c : T \Rightarrow^l T^{cl} \mid blame\ T\ T\ l^{cl} \mid \emptyset\ T^{cl}$
Cast Values $cv ::= cv1 \mid cv2$
 $cv1 ::= \emptyset\ T^{cl} : G \Rightarrow^l Dyn^{cl} \mid \emptyset\ T^{cl} : T_1 \rightarrow T_2 \Rightarrow^l T_3 \rightarrow T_4^{cl}$
 $\quad \mid cv1 : G \Rightarrow^l Dyn^{cl} \mid cv1 : T_1 \rightarrow T_2 \Rightarrow^l T_3 \rightarrow T_4^{cl}$
 $cv2 ::= blame\ T\ T\ l^{cl} \mid \emptyset\ T^{cl}$

$c \longrightarrow_{\cap IC} c$ Evaluation

Push blame to top level

$$\frac{}{blame\ T_I\ T_F\ l_1^{cl_1} : T_1 \Rightarrow^{l_2} T_2^{cl_2} \longrightarrow_{\cap IC} blame\ T_I\ T_2\ l_1^{cl_1}} \text{E-PushBlameIC}$$

Evaluate inside casts

$$\frac{\neg(is\ cast\ value\ c) \quad c \longrightarrow_{\cap IC} c'}{c : T_1 \Rightarrow^l T_2^{cl} \longrightarrow_{\cap IC} c' : T_1 \Rightarrow^l T_2^{cl}} \text{E-EvaluateIC}$$

Detect success or failure of casts

$$\frac{is\ cast\ value\ 1\ c \vee is\ empty\ cast\ c}{c : T \Rightarrow^l T^{cl} \longrightarrow_{\cap IC} c} \text{E-IdentityIC}$$

$$\frac{is\ cast\ value\ 1\ c \vee is\ empty\ cast\ c}{c : G \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G^{cl_2} \longrightarrow_{\cap IC} c} \text{E-SucceedIC}$$

$$\frac{is\ cast\ value\ 1\ c \vee is\ empty\ cast\ c \quad \neg(same\ ground\ G_1\ G_2) \quad initialType(c) = T_I}{c : G_1 \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G_2^{cl_2} \longrightarrow_{\cap IC} blame\ T_I\ G_2\ l_2^{cl_1}} \text{E-FailIC}$$

Mediate the transition between the two disciplines

$$\frac{is\ cast\ value\ 1\ c \vee is\ empty\ cast\ c \quad G\ is\ ground\ type\ of\ T \quad \neg(ground\ T)}{c : T \Rightarrow^l Dyn^{cl} \longrightarrow_{\cap IC} c : T \Rightarrow^l G^{cl} : G \Rightarrow^l Dyn^{cl}} \text{E-GroundIC}$$

$$\frac{is\ cast\ value\ 1\ c \vee is\ empty\ cast\ c \quad G\ is\ ground\ type\ of\ T \quad \neg(ground\ T)}{c : Dyn \Rightarrow^l T^{cl} \longrightarrow_{\cap IC} c : Dyn \Rightarrow^l G^{cl} : G \Rightarrow^l T^{cl}} \text{E-ExpandIC}$$

Figure 7: Intersection Casts Operational Semantics ($\longrightarrow_{\cap IC}$)

Syntax

Types $T ::= \text{Int} \mid \text{Bool} \mid \text{Dyn} \mid T \rightarrow T \mid T \cap \dots \cap T$
 Expressions $e ::= x \mid \lambda x : T . e \mid e e \mid n \mid \text{true} \mid \text{false} \mid e : c \cap \dots \cap c \mid \text{blame}_T l$
 Values $v ::= x \mid \lambda x : T . e \mid n \mid \text{true} \mid \text{false} \mid \text{blame}_T l \mid v : cv_1 \cap \dots \cap cv_n$ such that
 $\neg(\forall i \in 1..n . cv_i = \text{blame } T \text{ } l \text{ } l^{cl}) \wedge \neg(\forall i \in 1..n . cv_i = \emptyset \text{ } T^{cl})$

$e \rightarrow_{\cap CC} e$ Evaluation

Push blame to top level

$$\frac{\Gamma \vdash_{\cap CC} (\text{blame}_{T_2} l) e_2 : T_1}{(\text{blame}_{T_2} l) e_2 \rightarrow_{\cap CC} \text{blame}_{T_1} l} \text{E-PUSHBLAME1}$$

$$\frac{\Gamma \vdash_{\cap CC} e_1 (\text{blame}_{T_2} l) : T_1}{e_1 (\text{blame}_{T_2} l) \rightarrow_{\cap CC} \text{blame}_{T_1} l} \text{E-PUSHBLAME2}$$

$$\frac{\vdash_{\cap IC} c_1 : T_1 \dots \vdash_{\cap IC} c_n : T_n}{\text{blame}_T l : c_1 \cap \dots \cap c_n \rightarrow_{\cap CC} \text{blame}_{T_1 \cap \dots \cap T_n} l} \text{E-PUSHBLAMECAST}$$

Evaluate expressions

$$\frac{}{(\lambda x : T_1 \cap \dots \cap T_n . e) v \rightarrow_{\cap CC} [x \mapsto v]e} \text{E-APPABS} \quad \frac{e_1 \rightarrow_{\cap CC} e'_1}{e_1 e_2 \rightarrow_{\cap CC} e'_1 e_2} \text{E-APP1}$$

$$\frac{e_2 \rightarrow_{\cap CC} e'_2}{v_1 e_2 \rightarrow_{\cap CC} v_1 e'_2} \text{E-APP2} \quad \frac{e \rightarrow_{\cap CC} e'}{e : c_1 \cap \dots \cap c_n \rightarrow_{\cap CC} e' : c_1 \cap \dots \cap c_n} \text{E-EVALUATE}$$

Simulate casts on data types

$$\frac{\begin{array}{l} \text{is value } (v_1 : cv_1 \cap \dots \cap cv_n) \quad \exists i \in 1..n . \text{isArrowCompatible}(cv_i) \\ ((c_{11}, c_{12}, c_1^s), \dots, (c_{m1}, c_{m2}, c_m^s)) = \text{simulateArrow}(cv_1, \dots, cv_n) \end{array}}{(v_1 : cv_1 \cap \dots \cap cv_n) v_2 \rightarrow_{\cap CC} (v_1 : c_1^s \cap \dots \cap c_m^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2}} \text{E-SIMULATEARROW}$$

Merge casts

$$\frac{\begin{array}{l} \text{is value } (v : cv_1 \cap \dots \cap cv_n) \\ v : c_1'' \cap \dots \cap c_j'' = \text{mergeCasts}(v : cv_1 \cap \dots \cap cv_n : c_1' \cap \dots \cap c_m') \end{array}}{v : cv_1 \cap \dots \cap cv_n : c_1' \cap \dots \cap c_m' \rightarrow_{\cap CC} v : c_1'' \cap \dots \cap c_j''} \text{E-MERGECASTS}$$

Evaluate intersection casts

$$\frac{\neg(\forall i \in 1..n . \text{is cast value } c_i) \quad c_1 \rightarrow_{\cap IC} cv_1 \dots c_n \rightarrow_{\cap IC} cv_n}{v : c_1 \cap \dots \cap c_n \rightarrow_{\cap CC} v : cv_1 \cap \dots \cap cv_n} \text{E-EVALUATECASTS}$$

Transition from cast values to values

$$\frac{}{v : \text{blame } I_1 \text{ } F_1 \text{ } l_1^{cl_1} \cap \dots \cap \text{blame } I_n \text{ } F_n \text{ } l_n^{cl_n} \rightarrow_{\cap CC} \text{blame}_{(F_1 \cap \dots \cap F_n)} l_1} \text{E-PROPAGATEBLAME}$$

$$\frac{}{v : \emptyset \text{ } T_1^{cl_1} \cap \dots \cap \emptyset \text{ } T_n^{cl_n} \rightarrow_{\cap CC} v} \text{E-REMOVEEMPTY}$$

Figure 8: Cast Calculus Operational Semantics ($\rightarrow_{\cap CC}$)

$$\langle c \rangle^{cl} = c$$

$$\langle c : T_1 \Rightarrow^l T_2^{cl} \rangle^{cl'} = \langle c \rangle^{cl'} : T_1 \Rightarrow^l T_2^{cl'}$$

$$\langle \text{blame } T_I \ T_F \ l^{cl'} \rangle^{cl} = \text{blame } T_I \ T_F \ l^{cl}$$

$$\langle \emptyset \ T^{cl'} \rangle^{cl} = \emptyset \ T^{cl}$$

$$\text{isArrowCompatible}(c) = \text{Bool}$$

$$\text{isArrowCompatible}(c : T_{11} \rightarrow T_{12} \Rightarrow^l T_{21} \rightarrow T_{22}^{cl}) = \text{isArrowCompatible}(c)$$

$$\text{isArrowCompatible}(\emptyset \ (T_1 \rightarrow T_2)^{cl}) = \text{True}$$

$$\text{separateIntersectionCast}(c) = (c, c)$$

$$\text{separateIntersectionCast}(c : T_1 \Rightarrow^l T_2^{cl}) = (\emptyset \ T_1^{cl} : T_1 \Rightarrow^l T_2^{cl}, c)$$

$$\text{separateIntersectionCast}(\emptyset \ T^{cl}) = (\emptyset \ T^{cl}, \emptyset \ T^{cl})$$

$$\text{breakdownArrowType}(c) = (c, c)$$

$$\text{breakdownArrowType}(\emptyset \ T_{11} \rightarrow T_{12}^{cl} : T_{11} \rightarrow T_{12} \Rightarrow^l T_{21} \rightarrow T_{22}^{cl}) = (\emptyset \ T_{21}^{cl} : T_{21} \Rightarrow^l T_{11}^{cl}, \emptyset \ T_{12}^{cl} : T_{12} \Rightarrow^l T_{22}^{cl})$$

$$\text{breakdownArrowType}(\emptyset \ T_1 \rightarrow T_2^{cl}) = (\emptyset \ T_1^{cl}, \emptyset \ T_2^{cl})$$

$$\text{simulateArrow}(c_1, \dots, c_n) = ((c_{11}, c_{12}, c_1^s), \dots, (c_{m1}, c_{m2}, c_m^s))$$

$$\begin{aligned} (c'_1, \dots, c'_m) &= \text{filter } \text{isArrowCompatible} \ (c_1, \dots, c_n) \\ ((c_1^f, c_1^s), \dots, (c_m^f, c_m^s)) &= \text{map } \text{separateIntersectionCast} \ (\langle c'_1 \rangle^0, \dots, \langle c'_m \rangle^0) \\ ((c_{11}, c_{12}), \dots, (c_{m1}, c_{m2})) &= \text{map } \text{breakdownArrowType} \ (\langle c'_1 \rangle^1, \dots, \langle c'_m \rangle^m) \\ \hline \text{simulateArrow}(c_1, \dots, c_n) &= ((c_{11}, c_{12}, c_1^s), \dots, (c_{m1}, c_{m2}, c_m^s)) \end{aligned}$$

Figure 9: Definitions for auxiliary semantic functions

$$\boxed{\text{getCastLabel}(c) = cl}$$

$$\text{getCastLabel}(c : T_1 \Rightarrow^l T_2^{cl}) = cl$$

$$\text{getCastLabel}(\text{blame } T_I \ T_F \ l^{cl}) = cl$$

$$\text{getCastLabel}(\emptyset \ T^{cl}) = cl$$

$$\boxed{\text{sameCastLabel}(c, c) = \text{Bool}}$$

$$\text{sameCastLabel}(c_1, c_2) = \text{getCastLabel}(c_1) == 0$$

$$\text{sameCastLabel}(c_1, c_2) = \text{getCastLabel}(c_2) == 0$$

$$\text{sameCastLabel}(c_1, c_2) = \text{getCastLabel}(c_1) == \text{getCastLabel}(c_2)$$

$$\boxed{\text{joinCasts}(c, c) = c}$$

$$\text{joinCasts}(c : T_1 \Rightarrow^l T_2^{cl}, c') = \text{joinCasts}(c, c') : T_1 \Rightarrow^l T_2^{cl}$$

$$\text{joinCasts}(\text{blame } T_I \ T_F \ l^{cl}, c) = \text{blame } T_I \ T_F \ l^{cl}$$

$$\text{getCastLabel}(\emptyset \ T^{cl}, c) = \langle c \rangle^{cl}$$

$$\boxed{\text{mergeCasts}(e) = e}$$

$$\frac{(c'_1, \dots, c'_o) = [\text{joinCast } y \ x \mid x \leftarrow (c_{11}, \dots, c_{1m}), \ y \leftarrow (c_{21}, \dots, c_{2n}), \\ \text{sameCastLabel } y \ x \ \&\& \ \text{initialType}(y) == \text{finalType}(x)]}{\text{mergeCasts}(e : c_{11} \cap \dots \cap c_{1m} : c_{21} \cap \dots \cap c_{2n}) = e : c'_1 \cap \dots \cap c'_o}$$

Figure 10: Definitions for auxiliary semantic functions

2 Proofs

Lemma 1 (Consistency reduces to equality when comparing static types). *If T_1 and T_2 are static types then $T_1 = T_2 \iff T_1 \sim T_2$.*

Proof. We proceed by structural induction on T_1 .

Base cases:

- $T_1 = \text{Int}$.
 - If $\text{Int} = \text{Int}$ then, by the definition of \sim , $\text{Int} \sim \text{Int}$.
 - If $\text{Int} \sim \text{Int}$, then $\text{Int} = \text{Int}$.
- $T_1 = \text{Bool}$.
 - If $\text{Bool} = \text{Bool}$ then, by the definition of \sim , $\text{Bool} \sim \text{Bool}$.
 - If $\text{Bool} \sim \text{Bool}$, then $\text{Bool} = \text{Bool}$.

Induction step:

- $T_1 = T_{11} \rightarrow T_{12}$.
 - If $T_{11} \rightarrow T_{12} = T_{21} \rightarrow T_{22}$, for some T_{21} and T_{22} , then $T_{11} = T_{21}$ and $T_{12} = T_{22}$. By the induction hypothesis, $T_{11} \sim T_{21}$ and $T_{12} \sim T_{22}$. Therefore, by the definition of \sim , $T_{11} \rightarrow T_{12} \sim T_{21} \rightarrow T_{22}$.
 - If $T_{11} \rightarrow T_{12} \sim T_2$, then by the definition of \sim , $T_2 = T_{21} \rightarrow T_{22}$ and $T_{11} \sim T_{21}$ and $T_{12} \sim T_{22}$. By the induction hypothesis, $T_{11} = T_{21}$ and $T_{12} = T_{22}$. Therefore, $T_{11} \rightarrow T_{12} = T_{21} \rightarrow T_{22}$.
- $T_1 = T_{11} \cap \dots \cap T_{1n}$.
 - If $T_{11} \cap \dots \cap T_{1n} = T_2$, then $\exists T_{21} \dots T_{2n} . T_2 = T_{21} \cap \dots \cap T_{2n}$ and $T_{11} = T_{21}$ and ... and $T_{1n} = T_{2n}$. By the induction hypothesis, $T_{11} \sim T_{21}$ and ... and $T_{1n} \sim T_{2n}$. Therefore, by the definition of \sim , $T_{11} \cap \dots \cap T_{1n} \sim T_{21} \cap \dots \cap T_{2n}$.
 - If $T_{11} \cap \dots \cap T_{1n} \sim T_2$, then either:
 - * $\exists T_{21} \dots T_{2n} . T_2 = T_{21} \cap \dots \cap T_{2n}$ and $T_{11} \sim T_{21}$ and ... and $T_{1n} \sim T_{2n}$. By the induction hypothesis, $T_{11} = T_{21}$ and ... and $T_{1n} = T_{2n}$. Therefore, $T_{11} \cap \dots \cap T_{1n} = T_{21} \cap \dots \cap T_{2n}$.
 - * $T_{11} \sim T_2$ and ... and $T_{1n} \sim T_2$. By the induction hypothesis, $T_{11} = T_2$ and ... and $T_{1n} = T_2$. As $T_2 \cap \dots \cap T_2 = T_2$, then $T_{11} \cap \dots \cap T_{1n} = T_2$.

□

Theorem 1 (Conservative Extension). *Depends on Lemma 1. If e is fully static and T is a static type, then $\Gamma \vdash_{\cap S} e : T \iff \Gamma \vdash_{\cap G} e : T$.*

Proof. We proceed by induction on the length of the derivation tree of $\vdash_{\cap S}$ and $\vdash_{\cap G}$ for the right and left direction of the implication, respectively.

Base cases:

- Rule T-Var.
 - If $\Gamma \vdash_{\cap S} x : T$, then $x : T \in \Gamma$. Therefore, $\Gamma \vdash_{\cap G} x : T$.
 - If $\Gamma \vdash_{\cap G} x : T$, then $x : T \in \Gamma$. Therefore, $\Gamma \vdash_{\cap S} e : T$.
- Rule T-Int.
 - If $\Gamma \vdash_{\cap S} n : Int$, then $\Gamma \vdash_{\cap G} n : Int$.
 - If $\Gamma \vdash_{\cap G} n : Int$, then $\Gamma \vdash_{\cap S} n : Int$.
- Rule T-True.
 - If $\Gamma \vdash_{\cap S} true : Bool$, then $\Gamma \vdash_{\cap G} true : Bool$.
 - If $\Gamma \vdash_{\cap G} true : Bool$, then $\Gamma \vdash_{\cap S} true : Bool$.
- Rule T-False.
 - If $\Gamma \vdash_{\cap S} false : Bool$, then $\Gamma \vdash_{\cap G} false : Bool$.
 - If $\Gamma \vdash_{\cap G} false : Bool$, then $\Gamma \vdash_{\cap S} false : Bool$.

Induction step:

- Rule T-Abs.
 - If $\Gamma \vdash_{\cap S} \lambda x . T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$, then $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap S} e : T$. By the induction hypothesis, $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap G} e : T$. Therefore, $\Gamma \vdash_{\cap G} \lambda x . T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$.
 - If $\Gamma \vdash_{\cap G} \lambda x . T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$, then $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap G} e : T$. By the induction hypothesis, $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap S} e : T$. Therefore, $\Gamma \vdash_{\cap S} \lambda x . T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$.
- Rule T-Abs'.
 - If $\Gamma \vdash_{\cap S} \lambda x . T_1 \cap \dots \cap T_n . e : T_i \rightarrow T$, then $\Gamma, x : T_i \vdash_{\cap S} e : T$. By the induction hypothesis, $\Gamma, x : T_i \vdash_{\cap G} e : T$. Therefore, $\Gamma \vdash_{\cap G} \lambda x . T_1 \cap \dots \cap T_n . e : T_i \rightarrow T$.
 - If $\Gamma \vdash_{\cap G} \lambda x . T_1 \cap \dots \cap T_n . e : T_i \rightarrow T$, then $\Gamma, x : T_i \vdash_{\cap G} e : T$. By the induction hypothesis, $\Gamma, x : T_i \vdash_{\cap S} e : T$. Therefore, $\Gamma \vdash_{\cap S} \lambda x . T_1 \cap \dots \cap T_n . e : T_i \rightarrow T$.
- Rule T-App.
 - If $\Gamma \vdash_{\cap S} e_1 e_2 : T$ then $\Gamma \vdash_{\cap S} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\cap S} e_2 : T_1 \cap \dots \cap T_n$. By the induction hypothesis, $\Gamma \vdash_{\cap G} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\cap G} e_2 : T_1 \cap \dots \cap T_n$. By the definition of \triangleright , $T_1 \cap \dots \cap T_n \rightarrow T \triangleright T_1 \cap \dots \cap T_n \rightarrow T$. By the definition of \sim , $T_1 \cap \dots \cap T_n \sim T_1 \cap \dots \cap T_n$. Therefore, $\Gamma \vdash_{\cap G} e_1 e_2 : T$.
 - If $\Gamma \vdash_{\cap G} e_1 e_2 : T$ then $\Gamma \vdash_{\cap G} e_1 : PM$, $PM \triangleright T_1 \cap \dots \cap T_n \rightarrow T$, $\Gamma \vdash_{\cap G} e_2 : T'_1 \cap \dots \cap T'_n$ and $T'_1 \cap \dots \cap T'_n \sim T_1 \cap \dots \cap T_n$. By the definition of \triangleright , $PM = T_1 \cap \dots \cap T_n \rightarrow T$, therefore $\Gamma \vdash_{\cap G} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$. By Lemma 1, $T'_1 \cap \dots \cap T'_n = T_1 \cap \dots \cap T_n$, and therefore $\Gamma \vdash_{\cap G} e_2 : T_1 \cap \dots \cap T_n$. By the induction hypothesis, $\Gamma \vdash_{\cap S} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\cap S} e_2 : T_1 \cap \dots \cap T_n$. Therefore, $\Gamma \vdash_{\cap S} e_1 e_2 : T$.
- Rule T-Gen.

- If $\Gamma \vdash_{\cap S} e : T_1 \cap \dots \cap T_n$ then $\Gamma \vdash_{\cap S} e : T_1$ and ... and $\Gamma \vdash_{\cap S} e : T_n$. By the induction hypothesis, $\Gamma \vdash_{\cap G} e : T_1$ and ... and $\Gamma \vdash_{\cap G} e : T_n$. Therefore, $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$.
- If $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$ then $\Gamma \vdash_{\cap G} e : T_1$ and ... and $\Gamma \vdash_{\cap G} e : T_n$. By the induction hypothesis, $\Gamma \vdash_{\cap S} e : T_1$ and ... and $\Gamma \vdash_{\cap S} e : T_n$. Therefore $\Gamma \vdash_{\cap S} e : T_1 \cap \dots \cap T_n$.

- Rule T-Inst.

- If $\Gamma \vdash_{\cap S} e : T_i$ then $\Gamma \vdash_{\cap S} e : T_1 \cap \dots \cap T_n$, such that $T_i \in \{T_1, \dots, T_n\}$. By the induction hypothesis, $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$. As $T_i \in \{T_1, \dots, T_n\}$, then $\Gamma \vdash_{\cap G} e : T_i$.
- If $\Gamma \vdash_{\cap G} e : T_i$ then $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$, such that $T_i \in \{T_1, \dots, T_n\}$. By the induction hypothesis, $\Gamma \vdash_{\cap S} e : T_1 \cap \dots \cap T_n$. As $T_i \in \{T_1, \dots, T_n\}$, then $\Gamma \vdash_{\cap S} e : T_i$.

□

Theorem 2 (Monotonicity w.r.t. precision). *If $\Gamma \vdash_{\cap G} e : T$ and $e' \sqsubseteq e$ then $\Gamma \vdash_{\cap G} e' : T'$ and $T' \sqsubseteq T$.*

Proof. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_{\cap G} e : T$.

Base cases:

- Rule T-Var. If $\Gamma \vdash_{\cap G} x : T$ and $x \sqsubseteq x$, then $\Gamma \vdash_{\cap G} x : T$ and $T \sqsubseteq T$.
- Rule T-Int. If $\Gamma \vdash_{\cap G} n : Int$ and $n \sqsubseteq n$, then $\Gamma \vdash_{\cap G} n : Int$ and $Int \sqsubseteq Int$.
- Rule T-True. If $\Gamma \vdash_{\cap G} true : Bool$ and $true \sqsubseteq true$, then $\Gamma \vdash_{\cap G} true : Bool$ and $Bool \sqsubseteq Bool$.
- Rule T-False. If $\Gamma \vdash_{\cap G} false : Bool$ and $false \sqsubseteq false$, then $\Gamma \vdash_{\cap G} false : Bool$ and $Bool \sqsubseteq Bool$.

Induction step:

- Rule T-Abs. If $\Gamma \vdash_{\cap G} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$ and $\lambda x : T'_1 \cap \dots \cap T'_n . e' \sqsubseteq \lambda x : T_1 \cap \dots \cap T_n . e$, then by rule T-Abs, $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap G} e : T$, and by the definition of \sqsubseteq , $T'_1 \cap \dots \cap T'_n \sqsubseteq T_1 \cap \dots \cap T_n$ and $e' \sqsubseteq e$. By the induction hypothesis, $\Gamma, x : T'_1 \cap \dots \cap T'_n \vdash_{\cap G} e' : T'$ and $T' \sqsubseteq T$. By rule T-Abs, $\Gamma \vdash_{\cap G} \lambda x : T'_1 \cap \dots \cap T'_n . e' : T'_1 \cap \dots \cap T'_n \rightarrow T'$, and by the definition of \sqsubseteq , $T'_1 \cap \dots \cap T'_n \rightarrow T' \sqsubseteq T_1 \cap \dots \cap T_n \rightarrow T$.
- Rule T-Abs'. If $\Gamma \vdash_{\cap G} \lambda x : T_1 \cap \dots \cap T_n . e : T_i \rightarrow T$ and $\lambda x : T'_1 \cap \dots \cap T'_n . e' \sqsubseteq \lambda x : T_1 \cap \dots \cap T_n . e$, then by rule T-Abs', $\Gamma, x : T_i \vdash_{\cap G} e : T$, and by the definition of \sqsubseteq , $T'_1 \cap \dots \cap T'_n \sqsubseteq T_1 \cap \dots \cap T_n$ and $e' \sqsubseteq e$. By the induction hypothesis, $\Gamma, x : T'_i \vdash_{\cap G} e' : T'$ and $T' \sqsubseteq T$. By rule T-Abs', $\Gamma \vdash_{\cap G} \lambda x : T'_1 \cap \dots \cap T'_n . e' : T'_i \rightarrow T'$, and by the definition of \sqsubseteq , $T'_i \rightarrow T' \sqsubseteq T_i \rightarrow T$.
- Rule T-App. If $\Gamma \vdash_{\cap G} e_1 e_2 : T$ and $e'_1 e'_2 \sqsubseteq e_1 e_2$ then by rule T-App, $\Gamma \vdash_{\cap G} e_1 : PM$, $PM \triangleright T_{11} \cap \dots \cap T_{1n} \rightarrow T$, $\Gamma \vdash_{\cap G} e_2 : T_{21} \cap \dots \cap T_{2n}$, and $T_{21} \cap \dots \cap T_{2n} \sim T_{11} \cap \dots \cap T_{1n}$, and by the definition of \sqsubseteq , $e'_1 \sqsubseteq e_1$ and $e'_2 \sqsubseteq e_2$. By the induction hypothesis, $\Gamma \vdash_{\cap G} e'_1 : PM'$ and $PM' \sqsubseteq PM$ and $PM' \triangleright T'_{11} \cap \dots \cap T'_{1n} \rightarrow T'$ and $\Gamma \vdash_{\cap G} e'_2 : T'_{21} \cap \dots \cap T'_{2n}$ and $T'_{21} \cap \dots \cap T'_{2n} \sqsubseteq T_{21} \cap \dots \cap T_{2n}$ and $T'_{21} \cap \dots \cap T'_{2n} \sim T'_{11} \cap \dots \cap T'_{1n}$. By the definition of \sqsubseteq and \triangleright , $T'_{11} \cap \dots \cap T'_{1n} \rightarrow T' \sqsubseteq T_{11} \cap \dots \cap T_{1n} \rightarrow T$, and therefore, $T' \sqsubseteq T$. As $\Gamma \vdash_{\cap G} e'_1 e'_2 : T'$, it is proved.

- Rule T-Gen. If $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$ and $e' \sqsubseteq e$, then by rule T-Gen, $\Gamma \vdash_{\cap G} e : T_1$ and ... and $\Gamma \vdash_{\cap G} e : T_n$. By the induction hypothesis, $\Gamma \vdash_{\cap G} e' : T'_1$ and $T'_1 \sqsubseteq T_1$ and ... and $\Gamma \vdash_{\cap G} e' : T'_n$ and $T'_n \sqsubseteq T_n$. Then by rule T-Gen, $\Gamma \vdash_{\cap G} e' : T'_1 \cap \dots \cap T'_n$ and by the definition of \sqsubseteq , $T'_1 \cap \dots \cap T'_n \sqsubseteq T_1 \cap \dots \cap T_n$.
- Rule T-Inst. If $\Gamma \vdash_{\cap G} e : T_i$ and $e' \sqsubseteq e$, then by rule T-Inst, $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$ such that $T_i \in \{T_1, \dots, T_n\}$. By the induction hypothesis, $\Gamma \vdash_{\cap G} e' : T'_1 \cap \dots \cap T'_n$ and $T'_1 \cap \dots \cap T'_n \sqsubseteq T_1 \cap \dots \cap T_n$. Therefore, by rule T-Inst, $\Gamma \vdash_{\cap G} e' : T'_i$ and by the definition of \sqsubseteq , $T'_i \sqsubseteq T_i$.

□

Theorem 3 (Type preservation of cast insertion). *If $\Gamma \vdash_{\cap G} e : T$ then $\Gamma \vdash_{\cap CC} e \rightsquigarrow e' : T$ and $\Gamma \vdash_{\cap CC} e' : T$.*

Proof. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_{\cap G} e : T$.

Base cases:

- Rule T-Var. If $\Gamma \vdash_{\cap G} x : T$, then by rule T-Var, $x : T \in \Gamma$. By rule C-Var, $\Gamma \vdash_{\cap CC} x \rightsquigarrow x : T$ and by rule T-Var, $\Gamma \vdash_{\cap CC} x : T$.
- Rule T-Int. As $\Gamma \vdash_{\cap G} n : Int$, then by rule C-Int, $\Gamma \vdash_{\cap CC} n \rightsquigarrow n : Int$ and by rule T-Int, $\Gamma \vdash_{\cap CC} n : Int$.
- Rule T-True. As $\Gamma \vdash_{\cap G} true : Bool$, then by rule C-True, $\Gamma \vdash_{\cap CC} true \rightsquigarrow true : Bool$ and by rule T-True, $\Gamma \vdash_{\cap CC} true : Bool$.
- Rule T-False. As $\Gamma \vdash_{\cap G} false : Bool$, then by rule C-False, $\Gamma \vdash_{\cap CC} false \rightsquigarrow false : Bool$ and by rule T-False, $\Gamma \vdash_{\cap CC} false : Bool$, it is proved.

Induction step:

- Rule T-Abs. If $\Gamma \vdash_{\cap G} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$ then by rule T-Abs, $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap G} e : T$. By the induction hypothesis, $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap CC} e \rightsquigarrow e' : T$ and $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\cap CC} e' : T$. By rule C-Abs, $\Gamma \vdash_{\cap CC} \lambda x : T_1 \cap \dots \cap T_n . e \rightsquigarrow \lambda x : T_1 \cap \dots \cap T_n . e' : T_1 \cap \dots \cap T_n \rightarrow T$ and by rule T-Abs, $\Gamma \vdash_{\cap CC} \lambda x : T_1 \cap \dots \cap T_n . e' : T_1 \cap \dots \cap T_n \rightarrow T$.
- Rule T-Abs'. If $\Gamma \vdash_{\cap G} \lambda x : T_1 \cap \dots \cap T_n . e : T_i \rightarrow T$ then by rule T-Abs', $\Gamma, x : T_i \vdash_{\cap G} e : T$. By the induction hypothesis, $\Gamma, x : T_i \vdash_{\cap CC} e \rightsquigarrow e' : T$ and $\Gamma, x : T_i \vdash_{\cap CC} e' : T$. By rule C-Abs', $\Gamma \vdash_{\cap CC} \lambda x : T_1 \cap \dots \cap T_n . e \rightsquigarrow \lambda x : T_1 \cap \dots \cap T_n . e' : T_i \rightarrow T$ and by rule T-Abs', $\Gamma \vdash_{\cap CC} \lambda x : T_1 \cap \dots \cap T_n . e' : T_i \rightarrow T$.
- Rule T-App. If $\Gamma \vdash_{\cap G} e_1 e_2 : T$ then by rule T-App, $\Gamma \vdash_{\cap G} e_1 : PM, PM \triangleright T_1 \cap \dots \cap T_n \rightarrow T$, $\Gamma \vdash_{\cap G} e_2 : T'_1 \cap \dots \cap T'_n$ and $T'_1 \cap \dots \cap T'_n \sim T_1 \cap \dots \cap T_n$. By the induction hypothesis, $\Gamma \vdash_{\cap CC} e_1 \rightsquigarrow e'_1 : PM$ and $\Gamma \vdash_{\cap CC} e'_1 : PM$, and $\Gamma \vdash_{\cap CC} e_2 \rightsquigarrow e'_2 : T'_1 \cap \dots \cap T'_n$ and $\Gamma \vdash_{\cap CC} e'_2 : T'_1 \cap \dots \cap T'_n$. Therefore, by rule C-App, $\Gamma \vdash_{\cap CC} e_1 e_2 \rightsquigarrow e'_1 e'_2 : T$. By the definition of \preceq and S , $S, e \hookrightarrow e$, by rule T-IntersectionCast, $\Gamma \vdash_{\cap CC} e'_1 : T_1 \rightarrow T \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\cap CC} e'_2 : T_1 \cap \dots \cap T_n$. By rule T-App', $\Gamma \vdash_{\cap CC} e'_1 e'_2 : T \cap \dots \cap T$ and then by the properties of intersection types (modulo repetitions), $\Gamma \vdash_{\cap CC} e'_1 e'_2 : T$.

- Rule T-Gen. If $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$ then by rule T-Gen, $\Gamma \vdash_{\cap G} e : T_1$ and ... and $\Gamma \vdash_{\cap G} e : T_n$. By the induction hypothesis, $\Gamma \vdash_{\cap CC} e \rightsquigarrow e' : T_1$ and ... and $\Gamma \vdash_{\cap CC} e \rightsquigarrow e' : T_n$, and $\Gamma \vdash_{\cap CC} e' : T_1$ and ... and $\Gamma \vdash_{\cap CC} e' : T_n$. By rule C-Gen, $\Gamma \vdash_{\cap CC} e \rightsquigarrow e' : T_1 \cap \dots \cap T_n$ and by rule T-Gen, $\Gamma \vdash_{\cap CC} e' : T_1 \cap \dots \cap T_n$.
- Rule T-Inst. If $\Gamma \vdash_{\cap G} e : T_i$ then by rule T-Inst, $\Gamma \vdash_{\cap G} e : T_1 \cap \dots \cap T_n$, such that $T_i \in \{T_1, \dots, T_n\}$. By the induction hypothesis, $\Gamma \vdash_{\cap CC} e \rightsquigarrow e' : T_1 \cap \dots \cap T_n$ and $\Gamma \vdash_{\cap CC} e' : T_1 \cap \dots \cap T_n$. By rule C-Inst, $\Gamma \vdash_{\cap CC} e \rightsquigarrow e' : T_i$ and by rule T-Inst, $\Gamma \vdash_{\cap CC} e' : T_i$.

□

Theorem 4 (Monotonicity w.r.t precision of cast insertion). *If $\Gamma \vdash_{\cap CC} e_1 \rightsquigarrow e'_1 : T_1$ and $\Gamma \vdash_{\cap CC} e_2 \rightsquigarrow e'_2 : T_2$ and $e_1 \sqsubseteq e_2$ then $e'_1 \sqsubseteq e'_2$ and $T_1 \sqsubseteq T_2$.*

Proof. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_{\cap CC} e_1 \rightsquigarrow e'_1 : T$. Base cases:

- Rule C-Var. If $\Gamma \vdash_{\cap CC} x \rightsquigarrow x : T$ and $\Gamma \vdash_{\cap CC} x \rightsquigarrow x : T$, and $x \sqsubseteq x$, then $x \sqsubseteq x$ and $T \sqsubseteq T$.
- Rule C-Int. If $\Gamma \vdash_{\cap CC} n \rightsquigarrow n : Int$, $\Gamma \vdash_{\cap CC} n \rightsquigarrow n : Int$ and $n \sqsubseteq n$, then $n \sqsubseteq n$ and $Int \sqsubseteq Int$.
- Rule C-True. If $\Gamma \vdash_{\cap CC} true \rightsquigarrow true : Bool$, $\Gamma \vdash_{\cap CC} true \rightsquigarrow true : Bool$ and $true \sqsubseteq true$, then $true \sqsubseteq true$ and $Bool \sqsubseteq Bool$.
- Rule C-False. If $\Gamma \vdash_{\cap CC} false \rightsquigarrow false : Bool$, $\Gamma \vdash_{\cap CC} false \rightsquigarrow false : Bool$ and $false \sqsubseteq false$, then $false \sqsubseteq false$ and $Bool \sqsubseteq Bool$.

Induction step:

- Rule C-Abs. If $\Gamma \vdash_{\cap CC} \lambda x : T_{11} \cap \dots \cap T_{1n} . e_1 \rightsquigarrow \lambda x : T_{11} \cap \dots \cap T_{1n} . e'_1 : T_{11} \cap \dots \cap T_{1n} \rightarrow T_1$ and $\Gamma \vdash_{\cap CC} \lambda x : T_{21} \cap \dots \cap T_{2n} . e_2 \rightsquigarrow \lambda x : T_{21} \cap \dots \cap T_{2n} . e'_2 : T_{21} \cap \dots \cap T_{2n} \rightarrow T_2$ and $\lambda x : T_{11} \cap \dots \cap T_{1n} . e_1 \sqsubseteq \lambda x : T_{21} \cap \dots \cap T_{2n} . e_2$ then by rule C-Abs, $\Gamma, x : T_{11} \cap \dots \cap T_{1n} \vdash_{\cap CC} e_1 \rightsquigarrow e'_1 : T_1$ and $\Gamma, x : T_{21} \cap \dots \cap T_{2n} \vdash_{\cap CC} e_2 \rightsquigarrow e'_2 : T_2$ and by the definition of \sqsubseteq , $T_{11} \cap \dots \cap T_{1n} \sqsubseteq T_{21} \cap \dots \cap T_{2n}$ and $e_1 \sqsubseteq e_2$. By the induction hypothesis, $e'_1 \sqsubseteq e'_2$ and $T_1 \sqsubseteq T_2$. Therefore, by the definition of \sqsubseteq , $\lambda x : T_{11} \cap \dots \cap T_{1n} . e'_1 \sqsubseteq \lambda x : T_{21} \cap \dots \cap T_{2n} . e'_2$ and $T_{11} \cap \dots \cap T_{1n} \rightarrow T_1 \sqsubseteq T_{21} \cap \dots \cap T_{2n} \rightarrow T_2$.
- Rule C-Abs'. If $\Gamma \vdash_{\cap CC} \lambda x : T_{11} \cap \dots \cap T_{1n} . e_1 \rightsquigarrow \lambda x : T_{11} \cap \dots \cap T_{1n} . e'_1 : T_{1i} \rightarrow T_1$, such that $T_{1i} \in \{T_{11}, \dots, T_{1n}\}$, and $\Gamma \vdash_{\cap CC} \lambda x : T_{21} \cap \dots \cap T_{2n} . e_2 \rightsquigarrow \lambda x : T_{21} \cap \dots \cap T_{2n} . e'_2 : T_{2i} \rightarrow T_2$, such that $T_{2i} \in \{T_{21}, \dots, T_{2n}\}$, and $\lambda x : T_{11} \cap \dots \cap T_{1n} . e_1 \sqsubseteq \lambda x : T_{21} \cap \dots \cap T_{2n} . e_2$ then by the definition of C-Abs', $\Gamma, x : T_{1i} \vdash_{\cap CC} e_1 \rightsquigarrow e'_1 : T_1$ and $\Gamma, x : T_{2i} \vdash_{\cap CC} e_2 \rightsquigarrow e'_2 : T_2$ and by the definition of \sqsubseteq , $T_{11} \cap \dots \cap T_{1n} \sqsubseteq T_{21} \cap \dots \cap T_{2n}$ and $e_1 \sqsubseteq e_2$ and therefore $T_{1i} \sqsubseteq T_{2i}$. By the induction hypothesis, $e'_1 \sqsubseteq e'_2$ and $T_1 \sqsubseteq T_2$. Therefore, by the definition of \sqsubseteq , $\lambda x : T_{11} \cap \dots \cap T_{1n} . e'_1 \sqsubseteq \lambda x : T_{21} \cap \dots \cap T_{2n} . e'_2$ and $T_{1i} \rightarrow T_1 \sqsubseteq T_{2i} \rightarrow T_2$.
- Rule C-App. If $\Gamma \vdash_{\cap CC} e_{11} e_{12} \rightsquigarrow e''_{11} e''_{12} : T_1$ and $\Gamma \vdash_{\cap CC} e_{21} e_{22} \rightsquigarrow e''_{21} e''_{22} : T_2$ and $e_{11} e_{12} \sqsubseteq e_{21} e_{22}$ then by rule C-App, $\Gamma \vdash_{\cap CC} e_{11} \rightsquigarrow e'_{11} : PM_1$ and $PM_1 \triangleright T_{11} \cap \dots \cap T_{1n} \rightarrow T_1$ and $\Gamma \vdash_{\cap CC} e_{12} \rightsquigarrow e'_{12} : T'_{11} \cap \dots \cap T'_{1n}$ and $T'_{11} \cap \dots \cap T'_{1n} \sim T_{11} \cap \dots \cap T_{1n}$ and $PM_1 \preceq S_{11}$ and $T_{11} \cap \dots \cap T_{1n} \rightarrow T_1 \preceq S_{12}$ and $T'_{11} \cap \dots \cap T'_{1n} \preceq S_{13}$ and $T_{11} \cap \dots \cap T_{1n} \preceq S_{14}$ and $S_{11}, S_{12}, e'_{11} \hookrightarrow e''_{11}$ and $S_{13}, S_{14}, e'_{12} \hookrightarrow e''_{12}$ and $\Gamma \vdash_{\cap CC} e_{21} \rightsquigarrow e'_{21} : PM_2$ and $PM_2 \triangleright T_{21} \cap \dots \cap T_{2n} \rightarrow T_2$ and $\Gamma \vdash_{\cap CC} e_{22} \rightsquigarrow e'_{22} : T'_{21} \cap \dots \cap T'_{2n}$ and $T'_{21} \cap \dots \cap T'_{2n} \sim T_{21} \cap \dots \cap T_{2n}$ and $PM_2 \preceq S_{21}$ and $T_{21} \cap \dots \cap T_{2n} \rightarrow T_2 \preceq S_{22}$ and $T'_{21} \cap \dots \cap T'_{2n} \preceq S_{23}$ and $T_{21} \cap \dots \cap T_{2n} \preceq S_{24}$ and

$S_{21}, S_{22}, e'_{21} \hookrightarrow e''_{21}$ and $S_{23}, S_{24}, e'_{22} \hookrightarrow e''_{22}$. As, by the definition of \sqsubseteq , $e_{11} \sqsubseteq e_{21}$ and $e_{12} \sqsubseteq e_{22}$ then by the induction hypothesis, $e'_{11} \sqsubseteq e'_{21}$ and $PM_1 \sqsubseteq PM_2$ and $e'_{12} \sqsubseteq e'_{22}$ and $T'_{11} \cap \dots \cap T'_{1n} \sqsubseteq T'_{21} \cap \dots \cap T'_{2n}$. By the definition of \triangleright , we have that $PM_1 = T_{11} \cap \dots \cap T_{1n} \rightarrow T_1$ and $PM_2 = T_{21} \cap \dots \cap T_{2n} \rightarrow T_2$ and so $T_{11} \cap \dots \cap T_{1n} \rightarrow T_1 \sqsubseteq T_{21} \cap \dots \cap T_{2n} \rightarrow T_2$ and therefore by the definition of \sqsubseteq , $T_1 \sqsubseteq T_2$. As by the definition of \preceq , $S, S, e \hookrightarrow e$ and \sqsubseteq , $e'_{11} \sqsubseteq e'_{21}$ and $e'_{12} \sqsubseteq e'_{22}$, then by the definition of \sqsubseteq , $e'_{11} e'_{12} \sqsubseteq e'_{21} e'_{22}$ and $T_1 \sqsubseteq T_2$.

- Rule C-Gen. If $\Gamma \vdash_{\text{C-Gen}} e_1 \rightsquigarrow e'_1 : T_{11} \cap \dots \cap T_{1n}$ and $\Gamma \vdash_{\text{C-Gen}} e_2 \rightsquigarrow e'_2 : T_{21} \cap \dots \cap T_{2n}$ and $e_1 \sqsubseteq e_2$ then by rule C-Gen, $\Gamma \vdash_{\text{C-Gen}} e_1 \rightsquigarrow e'_1 : T_{11}$ and ... and $\Gamma \vdash_{\text{C-Gen}} e_1 \rightsquigarrow e'_1 : T_{1n}$ and $\Gamma \vdash_{\text{C-Gen}} e_2 \rightsquigarrow e'_2 : T_{21}$ and ... and $\Gamma \vdash_{\text{C-Gen}} e_2 \rightsquigarrow e'_2 : T_{2n}$. By the induction hypothesis, $e'_1 \sqsubseteq e'_2$ and $T_{11} \sqsubseteq T_{21}$ and ... and $T_{1n} \sqsubseteq T_{2n}$, and therefore by the definition of \sqsubseteq , $T_{11} \cap \dots \cap T_{1n} \sqsubseteq T_{21} \cap \dots \cap T_{2n}$.
- Rule C-Inst. If $\Gamma \vdash_{\text{C-Inst}} e_1 \rightsquigarrow e'_1 : T_{1i}$ and $\Gamma \vdash_{\text{C-Inst}} e_2 \rightsquigarrow e'_2 : T_{2i}$ and $e_1 \sqsubseteq e_2$ then by rule C-Inst, $\Gamma \vdash_{\text{C-Inst}} e_1 \rightsquigarrow e'_1 : T_{11} \cap \dots \cap T_{1n}$ and $\Gamma \vdash_{\text{C-Inst}} e_2 \rightsquigarrow e'_2 : T_{21} \cap \dots \cap T_{2n}$. By the induction hypothesis, $e'_1 \sqsubseteq e'_2$ and $T_{11} \cap \dots \cap T_{1n} \sqsubseteq T_{21} \cap \dots \cap T_{2n}$, and therefore, by the definition of \sqsubseteq , $T_{1i} \sqsubseteq T_{2i}$.

□

Corollary 1 (Monotonicity of cast insertion). *Corollary of Theorem 4. If $\Gamma \vdash_{\text{C-Gen}} e_1 \rightsquigarrow e'_1 : T_1$ and $\Gamma \vdash_{\text{C-Gen}} e_2 \rightsquigarrow e'_2 : T_2$ and $e_1 \sqsubseteq e_2$ then $e'_1 \sqsubseteq e'_2$.*

Theorem 5 (Conservative Extension). *If e is fully static, then $e \rightarrow_{\text{NS}} e' \iff e \rightarrow_{\text{CC}} e'$.*

Proof. We proceed by induction on the length of the derivation tree of \rightarrow_{NS} and \rightarrow_{CC} for the right and left direction of the implication, respectively. Base cases:

- Rule E-AppAbs. If $(\lambda x : T_1 \cap \dots \cap T_n . e) v \rightarrow_{\text{NS}} [x \mapsto v]e$ and $(\lambda x : T_1 \cap \dots \cap T_n . e) v \rightarrow_{\text{CC}} [x \mapsto v]e$, then it is proved.

Induction step:

- Rule E-App1.
 - If $e_1 e_2 \rightarrow_{\text{NS}} e'_1 e_2$ then by rule E-App1, $e_1 \rightarrow_{\text{NS}} e'_1$. By the induction hypothesis, $e_1 \rightarrow_{\text{CC}} e'_1$. Therefore, by rule E-App1, $e_1 e_2 \rightarrow_{\text{CC}} e'_1 e_2$
 - If $e_1 e_2 \rightarrow_{\text{CC}} e'_1 e_2$ then by rule E-App1, $e_1 \rightarrow_{\text{CC}} e'_1$. By the induction hypothesis, $e_1 \rightarrow_{\text{NS}} e'_1$. Therefore, by rule E-App1, $e_1 e_2 \rightarrow_{\text{NS}} e'_1 e_2$
- Rule E-App2.
 - If $v_1 e_2 \rightarrow_{\text{NS}} v_1 e'_2$ then by rule E-App2, $e_2 \rightarrow_{\text{NS}} e'_2$. By the induction hypothesis, $e_2 \rightarrow_{\text{CC}} e'_2$. Therefore, by rule E-App2, $v_1 e_2 \rightarrow_{\text{CC}} v_1 e'_2$
 - If $v_1 e_2 \rightarrow_{\text{CC}} v_1 e'_2$ then by rule E-App2, $e_2 \rightarrow_{\text{CC}} e'_2$. By the induction hypothesis, $e_2 \rightarrow_{\text{NS}} e'_2$. Therefore, by rule E-App2, $v_1 e_2 \rightarrow_{\text{NS}} v_1 e'_2$

□

Lemma 2 (Type preservation of \rightarrow_{IC}). *If $c \rightarrow_{\text{IC}} c'$ and*

- $\vdash_{\text{IC}} c : T$ *then $\vdash_{\text{IC}} c' : T$.*

- $initialType(c) = T$ then $initialType(c') = T$.

Proof. We proceed by induction on the length of the derivation tree of $\rightarrow_{\cap IC}$.

Base cases:

- Rule E-PushBlameIC.
 - If $\vdash_{\cap IC} blame\ T_I\ T_F\ l_1^{cl_1} : T_1 \Rightarrow^{l_2} T_2^{cl_2} : T_2$ and by rule E-PushBlameIC, $blame\ T_I\ T_F\ l_1^{cl_1} : T_1 \Rightarrow^{l_2} T_2^{cl_2} \rightarrow_{\cap IC} blame\ T_I\ T_2\ l_1^{cl_1} : T_2$, then by rule T-BlameIC, $\vdash_{\cap IC} blame\ T_I\ T_2\ l_1^{cl_1} : T_2$, then it is proved.
 - By the definition of $initialType$, $initialType(blame\ T_I\ T_F\ l_1^{cl_1} : T_1 \Rightarrow^{l_2} T_2^{cl_2}) = T_I$. By rule E-PushBlameIC, $blame\ T_I\ T_F\ l_1^{cl_1} : T_1 \Rightarrow^{l_2} T_2^{cl_2} \rightarrow_{\cap IC} blame\ T_I\ T_2\ l_1^{cl_1} : T_2$. Since $initialType(blame\ T_I\ T_2\ l_1^{cl_1}) = T_I$, it is proved.
- Rule E-IdentityIC.
 - If $\vdash_{\cap IC} c : T \Rightarrow^l T^{cl} : T$, then by rule T-SingleIC, $\vdash_{\cap IC} c : T$. By rule E-IdentityIC, $c : T \Rightarrow^l T^{cl} \rightarrow_{\cap IC} c$.
 - By the definitions of $initialType$, $initialType(c : T \Rightarrow^l T^{cl}) = initialType(c)$. By rule E-IdentityIC, $c : T \Rightarrow^l T^{cl} \rightarrow_{\cap IC} c$.
- Rule E-SucceedIC.
 - If $\vdash_{\cap IC} c : G \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G^{cl_2} : G$, then by rule T-SingleIC, $\vdash_{\cap IC} c : G$. By rule E-SucceedIC, $c : G \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G^{cl_2} \rightarrow_{\cap IC} c$.
 - Rule E-SucceedIC. By the definition of $initialType$, $initialType(c : G \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G^{cl_2}) = initialType(c)$. By rule E-SucceedIC, $c : G \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G^{cl_2} \rightarrow_{\cap IC} c$. Therefore it is proved.
- Rule E-FailIC.
 - If $\vdash_{\cap IC} c : G_1 \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G_2^{cl_2} : G_2$, and by rule E-FailIC, $c : G_1 \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G_2^{cl_2} \rightarrow_{\cap IC} blame\ T_I\ G_2\ l_2^{cl_1}$ then by rule T-BlameIC, $\vdash_{\cap IC} blame\ T_I\ G_2\ l_2^{cl_1} : G_2$.
 - By the definition of $initialType$, $initialType(c : G_1 \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G_2^{cl_2}) = T_I$. By rule E-FailIC, $c : G_1 \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G_2^{cl_2} \rightarrow_{\cap IC} blame\ T_I\ G_2\ l_2^{cl_1}$, then $initialType(blame\ T_I\ G_2\ l_2^{cl_1}) = T_I$.
- Rule E-GroundIC.
 - If $\vdash_{\cap IC} c : T \Rightarrow^l Dyn^{cl} : Dyn$ then by rule T-SingleIC, $\vdash_{\cap IC} c : T$. By rule E-GroundIC, $c : T \Rightarrow^l Dyn^{cl} \rightarrow_{\cap IC} c : T \Rightarrow^l G^{cl} : G \Rightarrow^l Dyn^{cl}$, then by rule T-SingleIC, $\vdash_{\cap IC} c : T \Rightarrow^l G^{cl} : G \Rightarrow^l Dyn^{cl} : Dyn$.
 - By the definition of $initialType$, $initialType(c : T \Rightarrow^l Dyn^{cl}) = initialType(c)$. By rule E-GroundIC, $c : T \Rightarrow^l Dyn^{cl} \rightarrow_{\cap IC} c : T \Rightarrow^l G^{cl} : G \Rightarrow^l Dyn^{cl}$, then $initialType(c : T \Rightarrow^l G^{cl} : G \Rightarrow^l Dyn^{cl}) = initialType(c)$.
- Rule E-ExpandIC.

- If $\vdash_{\cap IC} c : Dyn \Rightarrow^l T^{cl} : T$ then by rule T-SingleIC, $\vdash_{\cap IC} c : Dyn$. By rule E-ExpandIC, $c : Dyn \Rightarrow^l T^{cl} \longrightarrow_{\cap IC} c : Dyn \Rightarrow^l G^{cl} : G \Rightarrow^l T^{cl}$, then by rule T-SingleIC, $\vdash_{\cap IC} c : Dyn \Rightarrow^l G^{cl} : G \Rightarrow^l T^{cl} : T$.
- By the definition of *initialType*, $initialType(c : Dyn \Rightarrow^l T^{cl}) = initialType(c)$. By rule E-ExpandIC, $c : Dyn \Rightarrow^l T^{cl} \longrightarrow_{\cap IC} c : Dyn \Rightarrow^l G^{cl} : G \Rightarrow^l T^{cl}$. Since $initialType(c : Dyn \Rightarrow^l G^{cl} : G \Rightarrow^l T^{cl}) = initialType(c)$, it is proved.

Induction step:

- Rule E-EvaluateIC.

- If $\vdash_{\cap IC} c : T_1 \Rightarrow^l T_2^{cl} : T_2$ then by rule T-SingleIC, $\vdash_{\cap IC} c : T_1$. By rule E-EvaluateIC, $c \longrightarrow_{\cap IC} c'$. By the induction hypothesis, $\vdash_{\cap IC} c' : T_1$. By rule E-EvaluateIC, $c : T_1 \Rightarrow^l T_2^{cl} \longrightarrow_{\cap IC} c' : T_1 \Rightarrow^l T_2^{cl}$, then by rule T-SingleIC, $\vdash_{\cap IC} c' : T_1 \Rightarrow^l T_2^{cl} : T_2$.
- By the definition of *initialType*, $initialType(c : T_1 \Rightarrow^l T_2^{cl}) = initialType(c)$. By rule E-EvaluateIC, $c \longrightarrow_{\cap IC} c'$. By the induction hypothesis, $initialType(c') = initialType(c)$. By rule E-EvaluateIC, $c : T_1 \Rightarrow^l T_2^{cl} \longrightarrow_{\cap IC} c' : T_1 \Rightarrow^l T_2^{cl}$. Since $initialType(c' : T_1 \Rightarrow^l T_2^{cl}) = initialType(c')$, it is proved.

□

Lemma 3 (Progress of $\longrightarrow_{\cap IC}$). *If $\Gamma \vdash_{\cap IC} c : T$ and $initialType(c) = T_I$ then either c is a cast value or there exists a c' such that $c \longrightarrow_{\cap IC} c'$.*

Proof. We proceed by induction on the length of the derivation tree of $\vdash_{\cap IC} c : T$.

Base cases:

- Rule T-BlameIC. As $\vdash_{\cap IC} blame\ T_I\ T_F\ l^{cl} : T_F$, $initialType(blame\ T_I\ T_F\ l^{cl}) = T_I$ and $blame\ T_I\ T_F\ l^{cl}$ is a cast value, it is proved.
- Rule T-EmptyIC. As $\vdash_{\cap IC} \emptyset\ T^{cl} : T$, $initialType(\emptyset\ T^{cl}) = T$ and $\emptyset\ T^{cl}$ is a cast value, it is proved.

Induction step:

- Rule T-SingleIC. If $\vdash_{\cap IC} c : T_1 \Rightarrow^l T_2^{cl} : T_2$ and $initialType(c : T_1 \Rightarrow^l T_2^{cl}) = T_I$ then by rule T-SingleIC, $\vdash_{\cap IC} c : T_1$ and $initialType(c) = T_I$. By the induction hypothesis, either c is a cast value or there is a c' such that $c \longrightarrow_{\cap IC} c'$. If c is a cast value, then c can either be of the form $blame\ T_I\ T_F\ l^{cl}$, in which case by rule E-PushBlameIC, $blame\ T_I\ T_F\ l_1^{cl_1} : T_1 \Rightarrow^{l_2} T_2^{cl_2} \longrightarrow_{\cap IC} blame\ T_I\ T_2\ l_1^{cl_1}$ or c is a cast value 1 or is an empty cast. If c is a cast value 1 or is an empty cast then $c : T_1 \Rightarrow^l T_2^{cl}$ can be of one of the following forms:
 - $c : T \Rightarrow^l T^{cl}$. Then by rule E-IdentityIC, $c : T \Rightarrow^l T^{cl} \longrightarrow_{\cap IC} c$.
 - $c : G \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G^{cl_2}$. Then by rule E-SucceedIC, $c : G \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G^{cl_2} \longrightarrow_{\cap IC} c$.
 - $c : G_1 \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G_2^{cl_2}$. Then by rule E-FailIC, $c : G_1 \Rightarrow^{l_1} Dyn^{cl_1} : Dyn \Rightarrow^{l_2} G_2^{cl_2} \longrightarrow_{\cap IC} blame\ T_I\ G_2\ l_2^{cl_1}$.
 - $c : T \Rightarrow^l Dyn^{cl}$. Then by rule E-GroundIC, $c : T \Rightarrow^l Dyn^{cl} \longrightarrow_{\cap IC} c : T \Rightarrow^l G^{cl} : G \Rightarrow^l Dyn^{cl}$.

– $c : \text{Dyn} \Rightarrow^l T^{cl}$. Then by rule E-ExpandIC, $c : \text{Dyn} \Rightarrow^l T^{cl} \longrightarrow_{\text{NIC}} c : \text{Dyn} \Rightarrow^l G^{cl} : G \Rightarrow^l T^{cl}$.

If there is a c' such that $c \longrightarrow_{\text{NIC}} c'$, then by rule E-EvaluateIC, $c : T_1 \Rightarrow^l T_2^{cl} \longrightarrow_{\text{NIC}} c' : T_1 \Rightarrow^l T_2^{cl}$.

□

Lemma 4 (Type preservation of $\longrightarrow_{\text{NIC}}$). *Depends on Lemmas 2 and 3. If $\Gamma \vdash_{\text{NIC}} e : T_1 \cap \dots \cap T_n$ and $e \longrightarrow_{\text{NIC}} e'$ then $\Gamma \vdash_{\text{NIC}} e' : T_1 \cap \dots \cap T_m$ such that $m \leq n$.*

Proof. We proceed by induction on the length of the derivation tree of $\longrightarrow_{\text{NIC}}$.

Base cases:

- Rule E-PushBlame1. If $\Gamma \vdash_{\text{NIC}} \text{blame}_{T_2} l e_2 : T_1$ and $\text{blame}_{T_2} l e_2 \longrightarrow_{\text{NIC}} \text{blame}_{T_1} l$ then by rule T-Blame, $\Gamma \vdash_{\text{NIC}} \text{blame}_{T_1} l : T_1$.
- Rule E-PushBlame2. If $\Gamma \vdash_{\text{NIC}} e_1 \text{blame}_{T_2} l : T_1$ and $e_1 \text{blame}_{T_2} l \longrightarrow_{\text{NIC}} \text{blame}_{T_1} l$ then by rule T-Blame, $\Gamma \vdash_{\text{NIC}} \text{blame}_{T_1} l : T_1$.
- Rule E-PushBlameCast. If $\Gamma \vdash_{\text{NIC}} \text{blame}_T l : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n$ and $\text{blame}_T l : c_1 \cap \dots \cap c_n \longrightarrow_{\text{NIC}} \text{blame}_{T_1 \cap \dots \cap T_n} l$ then by rule T-Blame, $\Gamma \vdash_{\text{NIC}} \text{blame}_{T_1 \cap \dots \cap T_n} l : T_1 \cap \dots \cap T_n$.
- Rule E-AppAbs. There exists a type $T_1 \cap \dots \cap T_n$ such that we can deduce $\Gamma \vdash_{\text{NIC}} (\lambda x : T_1 \cap \dots \cap T_n . e) v : T$ from $\Gamma \vdash_{\text{NIC}} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\text{NIC}} v : T_1 \cap \dots \cap T_n$ (x does not occur in Γ). Moreover, $\Gamma \vdash_{\text{NIC}} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$ only if $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\text{NIC}} e : T$. By rule E-AppAbs, $(\lambda x : T_1 \cap \dots \cap T_n . e) v \longrightarrow_{\text{NIC}} [x \mapsto v]e$. To obtain $\Gamma \vdash_{\text{NIC}} [x \mapsto v]e : T$, it is sufficient to replace, in the proof of $\Gamma, x : T_1 \cap \dots \cap T_n \vdash_{\text{NIC}} e : T$, the statements $x : T_i$ (introduced by the rules T-Var and T-Inst) by the deductions of $\Gamma \vdash_{\text{NIC}} v : T_i$ for $1 \leq i \leq n$. (Proof adapted from [1])
- Rule E-SimulateArrow. If $\Gamma \vdash_{\text{NIC}} (v_1 : cv_1 \cap \dots \cap cv_n) v_2 : T_{12} \cap \dots \cap T_{n2}$, then by rule T-App', $\Gamma \vdash_{\text{NIC}} v_1 : cv_1 \cap \dots \cap cv_n : T_1 \cap \dots \cap T_n$ such that $\exists i \in 1..n . T_i = T_{i1} \rightarrow T_{i2}$ and $\Gamma \vdash_{\text{NIC}} v_2 : T_{11} \cap \dots \cap T_{n1}$. As $\Gamma \vdash_{\text{NIC}} v_1 : cv_1 \cap \dots \cap cv_n : T_1 \cap \dots \cap T_n$, then by rule T-IntersectionCast, $\Gamma \vdash_{\text{NIC}} v_1 : T_1'' \cap \dots \cap T_n''$ and $\vdash_{\text{NIC}} cv_1 : T_1$ and ... and $\vdash_{\text{NIC}} cv_n : T_n$ and $I_1 = \text{initialType}(cv_1)$ and ... and $I_n = \text{initialType}(cv_n)$ such that $\{I_1, \dots, I_n\} \subseteq \{T_1'', \dots, T_n''\}$ and $I_1 \cap \dots \cap I_n = T_1'' \cap \dots \cap T_n''$ and $n \leq l$. For the sake of simplicity let's elide cast labels and blame labels. By the definition of SimulateArrow, we have that $c_1' = c_1'' : T_{11}' \rightarrow T_{12}' \Rightarrow T_{11} \rightarrow T_{12}$ and ... and $c_m' = c_m'' : T_{m1}' \rightarrow T_{m2}' \Rightarrow T_{m1} \rightarrow T_{m2}$, for some $m \leq n$. Also, $c_{11} = \emptyset T_{11} : T_{11} \Rightarrow T_{11}'$ and ... and $c_{m1} = \emptyset T_{m1} : T_{m1} \Rightarrow T_{m1}'$ and $c_{12} : \emptyset T_{12}' : T_{12}' \Rightarrow T_{12}$ and ... and $c_{m2} = \emptyset T_{m2}' : T_{m2}' \Rightarrow T_{m2}$ and $\text{initialType}(c_1^s) = I_1$ and ... and $\text{initialType}(c_m^s) = I_m$ and $\vdash_{\text{NIC}} c_1^s : T_{11}' \rightarrow T_{12}'$ and ... and $\vdash_{\text{NIC}} c_m^s : T_{m1}' \rightarrow T_{m2}'$. As by rule T-Gen and T-Inst $\Gamma \vdash_{\text{NIC}} v_1 : T_1'' \cap \dots \cap T_m''$ and $I_1 \cap \dots \cap I_m = T_1'' \cap \dots \cap T_m''$, then by rule T-IntersectionCast, $\Gamma \vdash_{\text{NIC}} v_1 : c_1^s \cap \dots \cap c_m^s : T_{11}' \rightarrow T_{12}' \cap \dots \cap T_{m1}' \rightarrow T_{m2}'$. As by rule T-Gen and T-Inst $\Gamma \vdash_{\text{NIC}} v_2 : T_{11} \cap \dots \cap T_{m1}$ and $\vdash_{\text{NIC}} c_{11} : T_{11}'$ and ... and $\vdash_{\text{NIC}} c_{m1} : T_{m1}'$ and $\text{initialType}(c_{11}) = T_{11}$ and ... and $\text{initialType}(c_{m1}) = T_{m1}$, then by rule T-IntersectionCast, $\Gamma \vdash_{\text{NIC}} v_2 : c_{11} \cap \dots \cap c_{m1} : T_{11}' \cap \dots \cap T_{m1}'$. Therefore, by rule T-App', $\Gamma \vdash_{\text{NIC}} (v_1 : c_1^s \cap \dots \cap c_m^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : T_{12}' \cap \dots \cap T_{m2}'$. As $\vdash_{\text{NIC}} c_{12} : T_{12}$ and ... and $\vdash_{\text{NIC}} c_{m2} : T_{m2}$ and $\text{initialType}(c_{12}) = T_{12}'$ and ... and $\text{initialType}(c_{m2}) = T_{m2}'$, then by rule T-IntersectionCast, $\Gamma \vdash_{\text{NIC}} (v_1 : c_1^s \cap \dots \cap c_m^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2} : T_{12} \cap \dots \cap T_{m2}$.

By rule E-SimulateArrow, $(v_1 : cv_1 \cap \dots \cap cv_n) v_2 \rightarrow_{\cap CC}$
 $(v_1 : c_1^s \cap \dots \cap c_m^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2}$, therefore it is proved.

- Rule E-MergeCasts. If $\Gamma \vdash_{\cap CC} v : cv_1 \cap \dots \cap cv_n : c'_1 \cap \dots \cap c'_m : F'_1 \cap \dots \cap F'_m$ then by rule T-IntersectionCasts, $\Gamma \vdash_{\cap CC} v : cv_1 \cap \dots \cap cv_n : F_1 \cap \dots \cap F_n$ and $\vdash_{\cap IC} c'_1 : F'_1$ and ... and $\vdash_{\cap IC} c'_m : F'_m$ and $initialType(c'_1) = I'_1$ and $initialType(c'_m) = I'_m$ such that $\{I'_1, \dots, I'_m\} \subseteq \{F_1, \dots, F_n\}$ and $I'_1 \cap \dots \cap I'_m = F_1 \cap \dots \cap F_m$ and $m \leq n$. As $\Gamma \vdash_{\cap CC} v : cv_1 \cap \dots \cap cv_n : F_1 \cap \dots \cap F_n$ then by rule T-IntersectionCast, $\Gamma \vdash_{\cap CC} v : T_1 \cap \dots \cap T_l$ and $\vdash_{\cap IC} cv_1 : F_1$ and ... and $\vdash_{\cap IC} cv_n : F_n$ and $initialType(cv_1) : I_1$ and ... and $initialType(cv_n) : I_n$ such that $\{I_1, \dots, I_n\} \subseteq \{T_1, \dots, T_l\}$ and $I_1 \cap \dots \cap I_n = T_1 \cap \dots \cap T_n$ and $n \leq l$. By the definition of mergeCasts, $\vdash_{\cap IC} c'_1 : F''_1$ and ... and $\vdash_{\cap IC} c'_j : F''_j$ and $initialType(c'_1) = I''_1$ and ... and $initialType(c'_j) = I''_j$ such that $\{I''_1, \dots, I''_j\} \subseteq \{T_1, \dots, T_l\}$ and $I''_1 \cap \dots \cap I''_j = T_1 \cap \dots \cap T_j$ and $\{F''_1, \dots, F''_j\} \subseteq \{F'_1, \dots, F'_m\}$ and $F''_1 \cap \dots \cap F''_j = F'_1 \cap \dots \cap F'_j$ and $j \leq l$ and $j \leq m$. By rule T-Gen and T-Inst, $\Gamma \vdash_{\cap CC} v : T_1 \cap \dots \cap T_j$ and therefore by rule T-IntersectionCast, $\Gamma \vdash_{\cap CC} v : c'_1 \cap \dots \cap c'_j : F''_1 \cap \dots \cap F''_j$. By rule E-MergeCasts, $v : cv_1 \cap \dots \cap cv_n : c'_1 \cap \dots \cap c'_m \rightarrow_{\cap CC} v : c''_1 \cap \dots \cap c''_j$.
- Rule E-EvaluateCasts. If $\Gamma \vdash_{\cap CC} v : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n$ then by rule T-IntersectionCast, $\Gamma \vdash_{\cap CC} v : T'_1 \cap \dots \cap T'_n$ and $\vdash_{\cap IC} c_1 : T_1$ and ... and $\vdash_{\cap IC} c_n : T_n$ and $I_1 = initialType(c_1)$ and ... and $I_n = initialType(c_n)$ and $I_1 \cap \dots \cap I_n = T'_1 \cap \dots \cap T'_n$. By rule E-EvaluateCasts, $c_1 \rightarrow_{\cap IC} cv_1$ and ... and $c_n \rightarrow_{\cap IC} cv_n$. By Lemmas 2 and 3, $\vdash_{\cap IC} cv_1 : T_1$ and $initialType(cv_1) = I_1$ and ... and $\vdash_{\cap IC} cv_n : T_n$ and $initialType(cv_n) = I_n$. Therefore by rule T-IntersectionCast, $\Gamma \vdash_{\cap CC} v : cv_1 \cap \dots \cap cv_n : T_1 \cap \dots \cap T_n$. By rule E-EvaluateCasts, $v : c_1 \cap \dots \cap c_n \rightarrow_{\cap CC} v : cv_1 \cap \dots \cap cv_n$.
- Rule E-PropagateBlame. If $\Gamma \vdash_{\cap CC} v : blame T'_1 T_1 l_1^{m_1} \cap \dots \cap blame T'_n T_n l_n^{m_n} : T_1 \cap \dots \cap T_n$ and by rule E-PropagateBlame $v : blame T'_1 T_1 l_1^{m_1} \cap \dots \cap blame T'_n T_n l_n^{m_n} \rightarrow_{\cap CC} blame_{(T_1 \cap \dots \cap T_n)} l_1$, then by rule T-Blame, $\Gamma \vdash_{\cap CC} blame_{(T_1 \cap \dots \cap T_n)} l_1 : T_1 \cap \dots \cap T_n$.
- Rule E-RemoveEmpty. If $\Gamma \vdash_{\cap CC} v : \emptyset T_1^{m_1} \cap \dots \cap \emptyset T_n^{m_n} : T_1 \cap \dots \cap T_n$, then by rule T-IntersectionCast, $\Gamma \vdash_{\cap CC} v : T_1 \cap \dots \cap T_n$ and $\vdash_{\cap IC} \emptyset T_1^{m_1} : T_1$ and ... and $\vdash_{\cap IC} \emptyset T_n^{m_n} : T_n$ and $initialType(\emptyset T_1^{m_1}) = T_1$ and ... and $initialType(\emptyset T_n^{m_n}) = T_n$. Therefore, by rule E-RemoveEmpty, $v : \emptyset T_1^{m_1} \cap \dots \cap \emptyset T_n^{m_n} \rightarrow_{\cap CC} v$.

Induction step:

- Rule E-App1. There are two possibilities:
 - If $\Gamma \vdash_{\cap CC} e_1 e_2 : T$, then by rule T-App, $\Gamma \vdash_{\cap CC} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\cap CC} e_2 : T_1 \cap \dots \cap T_n$. By rule E-App1, $e_1 \rightarrow_{\cap IC} e'_1$, so by the induction hypothesis, $\Gamma \vdash_{\cap CC} e'_1 : T_1 \cap \dots \cap T_n \rightarrow T$. As by rule E-App1, $e_1 e_2 \rightarrow_{\cap IC} e'_1 e_2$, then by rule T-App, $\Gamma \vdash_{\cap CC} e'_1 e_2 : T$.
 - If $\Gamma \vdash_{\cap CC} e_1 e_2 : T_{12} \cap \dots \cap T_{n2}$, then by rule T-App', $\Gamma \vdash_{\cap CC} e_1 : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}$ and $\Gamma \vdash_{\cap CC} e_2 : T_{11} \cap \dots \cap T_{n1}$. By rule E-App1, $e_1 \rightarrow_{\cap IC} e'_1$, so by the induction hypothesis, $\Gamma \vdash_{\cap CC} e'_1 : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}$. As by rule E-App1, $e_1 e_2 \rightarrow_{\cap IC} e'_1 e_2$, then by rule T-App', $\Gamma \vdash_{\cap CC} e'_1 e_2 : T_{12} \cap \dots \cap T_{n2}$.
- Rule E-App2. There are two possibilities:
 - If $\Gamma \vdash_{\cap CC} v_1 e_2 : T$, then by rule T-App, $\Gamma \vdash_{\cap CC} v_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\cap CC} e_2 : T_1 \cap \dots \cap T_n$. By rule E-App2, $e_2 \rightarrow_{\cap IC} e'_2$, so by the induction hypothesis, $\Gamma \vdash_{\cap CC} e'_2 : T_1 \cap \dots \cap T_n$. As by rule E-App2, $v_1 e_2 \rightarrow_{\cap IC} v_1 e'_2$, then by rule T-App, $\Gamma \vdash_{\cap CC} v_1 e'_2 : T$.

- If $\Gamma \vdash_{\text{NCC}} v_1 e_2 : T_{12} \cap \dots \cap T_{n2}$, then by rule T-App', $\Gamma \vdash_{\text{NCC}} v_1 : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}$ and $\Gamma \vdash_{\text{NCC}} e_2 : T_{11} \cap \dots \cap T_{n1}$. By rule E-App2, $e_2 \rightarrow_{\text{NIC}} e'_2$, so by the induction hypothesis, $\Gamma \vdash_{\text{NCC}} e'_2 : T_{11} \cap \dots \cap T_{n1}$. As by rule E-App1, $v_1 e_2 \rightarrow_{\text{NIC}} v_1 e'_2$, then by rule T-App', $\Gamma \vdash_{\text{NCC}} v_1 e'_2 : T_{12} \cap \dots \cap T_{n2}$.
- Rule E-Evaluate. If $\Gamma \vdash_{\text{NCC}} e : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n$, then by rule T-IntersectionCast, $\Gamma \vdash_{\text{NCC}} e : T'_1 \cap \dots \cap T'_n$, $\vdash_{\text{NIC}} c_1 : T_1$ and ... and $\vdash_{\text{NIC}} c_n : T_n$ and $\text{initialType}(c_1) \cap \dots \cap \text{initialType}(c_n) = T'_1 \cap \dots \cap T'_n$. By rule E-Evaluate, $e \rightarrow_{\text{NIC}} e'$, so by the induction hypothesis, $\Gamma \vdash_{\text{NCC}} e' : T$. As by rule E-Evaluate, $e : c_1 \cap \dots \cap c_n \rightarrow_{\text{NIC}} e' : c_1 \cap \dots \cap c_n$, then by rule T-IntersectionCast, $\Gamma \vdash_{\text{NCC}} e' : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n$.

□

Lemma 5 (Progress of \rightarrow_{NCC}). *If $\Gamma \vdash_{\text{NCC}} e : T$ then either e is a value or there exists an e' such that $e \rightarrow_{\text{NCC}} e'$.*

Proof. We proceed by induction on the length of the derivation tree of $\Gamma \vdash_{\text{NCC}} e : T$.

Base cases:

- Rule T-Var. If $\Gamma \vdash_{\text{NCC}} x : T$, then x is a value.
- Rule T-Int. If $\Gamma \vdash_{\text{NCC}} n : \text{Int}$ then n is a value.
- Rule T-True. If $\Gamma \vdash_{\text{NCC}} \text{true} : \text{Bool}$ then true is a value.
- Rule T-False. If $\Gamma \vdash_{\text{NCC}} \text{false} : \text{Bool}$ then false is a value.

Induction step:

- Rule T-Abs. If $\Gamma \vdash_{\text{NCC}} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$ then $\lambda x : T_1 \cap \dots \cap T_n . e$ is a value.
- Rule T-Abs'. If $\Gamma \vdash_{\text{NCC}} \lambda x : T_1 \cap \dots \cap T_n . e : T_i \rightarrow T$ then $\lambda x : T_1 \cap \dots \cap T_n . e$ is a value.
- Rule T-App. If $\Gamma \vdash_{\text{NCC}} e_1 e_2 : T$ then by rule T-App, $\Gamma \vdash_{\text{NCC}} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\text{NCC}} e_2 : T_1 \cap \dots \cap T_n$. By the induction hypothesis, e_1 is either a value or there is a e'_1 such that $e_1 \rightarrow_{\text{NCC}} e'_1$ and e_2 is either a value or there is a e'_2 such that $e_2 \rightarrow_{\text{NCC}} e'_2$. If e_1 is a value, then by rule E-PushBlame1, $(\text{blame}_{T_2} l) e_2 \rightarrow_{\text{NCC}} \text{blame}_{T_1} l$. If e_2 is a value, then by rule E-PushBlame2, $e_1 (\text{blame}_{T_2} l) \rightarrow_{\text{NCC}} \text{blame}_{T_1} l$. If e_1 is not a value, then by rule E-App1, $e_1 e_2 \rightarrow_{\text{NCC}} e'_1 e_2$. If e_1 is a value and e_2 is not a value, then by rule E-App2, $v_1 e_2 \rightarrow_{\text{NCC}} v_1 e'_2$. If both e_1 and e_2 are values then e_1 must be an abstraction $(\lambda x : T_1 \cap \dots \cap T_n . e)$, and by rule E-AppAbs $(\lambda x : T_1 \cap \dots \cap T_n . e) v_2 \rightarrow_{\text{NCC}} [x \mapsto v_2]e$.
- Rule T-Gen. If $\Gamma \vdash_{\text{NCC}} e : T_1 \cap \dots \cap T_n$ then by rule T-Gen, $\Gamma \vdash_{\text{NCC}} e : T_1$ and ... and $\Gamma \vdash_{\text{NCC}} e : T_n$. By the induction hypothesis, either e is a value or there exists an e' such that $e \rightarrow_{\text{NCC}} e'$.
- Rule T-Inst. If $\Gamma \vdash_{\text{NCC}} e : T_i$ then by rule T-Inst, $\Gamma \vdash_{\text{NCC}} e : T_1 \cap \dots \cap T_n$, such that $T_i \in \{T_1, \dots, T_n\}$. By the induction hypothesis, either e is a value or there exists an e' such that $e \rightarrow_{\text{NCC}} e'$.

- Rule T-App'. If $\Gamma \vdash_{\text{NCC}} e_1 e_2 : T_{12} \cap \dots \cap T_{n2}$ then by rule T-App', $\Gamma \vdash_{\text{NCC}} e_1 : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}$ and $\Gamma \vdash_{\text{NCC}} e_2 : T_{11} \cap \dots \cap T_{n1}$. By the induction hypothesis, e_1 is either a value or there is a e'_1 such that $e_1 \rightarrow_{\text{NCC}} e'_1$ and e_2 is either a value or there is a e'_2 such that $e_2 \rightarrow_{\text{NCC}} e'_2$. If e_1 is a value, then by rule E-PushBlame1, $(\text{blame}_{T_2} l) e_2 \rightarrow_{\text{NCC}} \text{blame}_{T_1} l$. If e_2 is a value, then by rule E-PushBlame2, $e_1 (\text{blame}_{T_2} l) \rightarrow_{\text{NCC}} \text{blame}_{T_1} l$. If e_1 is not a value, then by rule E-App1, $e_1 e_2 \rightarrow_{\text{NCC}} e'_1 e_2$. If e_1 is a value and e_2 is not a value, then by rule E-App2, $v_1 e_2 \rightarrow_{\text{NCC}} v_1 e'_2$. If both e_1 and e_2 are values then e_1 must be an abstraction $(\lambda x : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}. e)$, and by rule E-AppAbs $(\lambda x : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}. e) v_2 \rightarrow_{\text{NCC}} [x \mapsto v_2]e$.
- Rule T-IntersectionCast. If $\Gamma \vdash_{\text{NCC}} e : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n$ then by rule T-IntersectionCast, $\Gamma \vdash_{\text{NCC}} e : T'_1 \cap \dots \cap T'_n$. By the induction hypothesis, e is either a value, or there is an e' such that $e \rightarrow_{\text{NCC}} e'$. If e is a value, then either by rule E-EvaluateCasts, $v : c_1 \cap \dots \cap c_n \rightarrow_{\text{NCC}} v : cv_1 \cap \dots \cap cv_n$, or by rule E-PushBlameCast, $\text{blame}_{T'_1 \cap \dots \cap T'_n} l : c_1 \cap \dots \cap c_n \rightarrow_{\text{NCC}} \text{blame}_{T_1 \cap \dots \cap T_n} l$. If there is an e' such that $e \rightarrow_{\text{NCC}} e'$, then by rule E-Evaluate, $e : c_1 \cap \dots \cap c_n \rightarrow_{\text{NCC}} e' : c_1 \cap \dots \cap c_n$.
- Rule T-Blame. If $\Gamma \vdash_{\text{NCC}} \text{blame}_T l : T$ then $\text{blame}_T l$ is a value.

□

Theorem 6 (Type Safety of \rightarrow_{NCC}). *Depends on Lemmas 4 and 5. Both Type Preservation and Progress hold for \rightarrow_{NCC} .*

Proof. We have Type Preservation (by Lemma 4) and Progress (by Lemma 5) for \rightarrow_{NCC} . □

Theorem 7 (Blame Theorem). *If $\Gamma \vdash_{\text{NCC}} e : T$ and $e \rightarrow_{\text{NCC}}^* \text{blame}_T l$ then l is not a safe cast of e .*

Lemma 6 (Gradual Guarantee for \rightarrow_{NIC}). *If $\vdash_{\text{NIC}} c_1 : T_1$ and $\vdash_{\text{NIC}} c_2 : T_2$ and $c_1 \sqsubseteq c_2$ then:*

1. *if $c_2 \rightarrow_{\text{NIC}} c'_2$ then $c_1 \rightarrow_{\text{NIC}}^* c'_1$ and $c'_1 \sqsubseteq c'_2$.*

Lemma 7 (Extra Cast on the Left). *If $\Gamma \vdash_{\text{NCC}} v : T$, $\Gamma \vdash_{\text{NCC}} v' : T'$, $\vdash_{\text{NIC}} c_1 : T_1$ and ... and $\vdash_{\text{NIC}} c_n : T_n$, $T_1 \cap \dots \cap T_n \sqsubseteq T'$, $\text{initialType}(c_1) \cap \dots \cap \text{initialType}(c_n) = T$ and $v \sqsubseteq v'$ then $v : c_1 \cap \dots \cap c_n \rightarrow_{\text{NCC}}^* v''$ and $v'' \sqsubseteq v'$.*

Lemma 8 (Catchup to Value on the Right). *If $\Gamma \vdash_{\text{NCC}} e : T$, $\Gamma \vdash_{\text{NCC}} v : T'$, $e \sqsubseteq v$, then $e \rightarrow_{\text{NCC}}^* e'$ and $e' \sqsubseteq v$.*

Lemma 9 (Simulation of Function Application). *Depends on Lemma 4. Suppose $\Gamma \vdash_{\text{NCC}} \lambda x : T'_1 \cap \dots \cap T'_n . e' : T'_1 \cap \dots \cap T'_n \rightarrow T'$, $\Gamma \vdash_{\text{NCC}} v' : T'_1 \cap \dots \cap T'_n$, either $\Gamma \vdash_{\text{NCC}} v_1 : T_1 \cap \dots \cap T_n \rightarrow T$ or $\Gamma \vdash_{\text{NCC}} v_1 : T_1 \rightarrow T \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\text{NCC}} v_2 : T_1 \cap \dots \cap T_n, T_1 \cap \dots \cap T_n \rightarrow T \sqsubseteq T'_1 \cap \dots \cap T'_n \rightarrow T'$. If $v_1 \sqsubseteq \lambda x : T'_1 \cap \dots \cap T'_n . e'$ and $v_2 \sqsubseteq v'$, then $v_1 v_2 \rightarrow_{\text{NCC}}^* e$ such that $e \sqsubseteq [x \mapsto v']e'$ and $\Gamma \vdash_{\text{NCC}} e : T$.*

Proof. We proceed by induction on the length of the derivation tree of $v_1 \sqsubseteq \lambda x : T'_1 \cap \dots \cap T'_n . e'$.

Base cases:

- $\lambda x : T_1 \cap \dots \cap T_n . e \sqsubseteq \lambda x : T'_1 \cap \dots \cap T'_n . e'$. As $\Gamma \vdash_{\text{NCC}} \lambda x : T_1 \cap \dots \cap T_n . e : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\text{NCC}} v_2 : T_1 \cap \dots \cap T_n$ then $\Gamma \vdash_{\text{NCC}} (\lambda x : T_1 \cap \dots \cap T_n . e) v_2 : T$. As $\lambda x : T_1 \cap \dots \cap T_n . e \sqsubseteq \lambda x : T'_1 \cap \dots \cap T'_n . e'$, then $e \sqsubseteq e'$. By Rule E-AppAbs, $(\lambda x : T_1 \cap \dots \cap T_n . e) v_2 \rightarrow_{\text{NCC}} [x \mapsto v_2]e$. By the definition of \sqsubseteq , $[x \mapsto v_2]e \sqsubseteq [x \mapsto v']e'$ and by Lemma 4, $\Gamma \vdash_{\text{NCC}} [x \mapsto v_2]e : T$.

Induction step:

- $v : cv_1 \cap \dots \cap cv_n \sqsubseteq \lambda x : T'_1 \cap \dots \cap T'_n . e'$. As $\Gamma \vdash_{\cap CC} v : cv_1 \cap \dots \cap cv_n : T_1 \rightarrow T \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\cap CC} v_2 : T_1 \cap \dots \cap T_n$ and $T_1 \rightarrow T \cap \dots \cap T_n \rightarrow T \sqsubseteq T'_1 \cap \dots \cap T'_n \rightarrow T'$ then $T \sqsubseteq T'$. As $v : cv_1 \cap \dots \cap cv_n \sqsubseteq \lambda x : T'_1 \cap \dots \cap T'_n . e'$, then $v \sqsubseteq \lambda x : T'_1 \cap \dots \cap T'_n . e'$. By rule E-SimulateArrow, $(v : cv_1 \cap \dots \cap cv_n) v_2 \rightarrow_{\cap CC} (v : c_1^s \cap \dots \cap c_m^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2}$, with $((c_{11}, c_{12}, c_1^s), \dots, (c_{m1}, c_{m2}, c_m^s)) = \text{simulateArrow}(cv_1, \dots, cv_n)$. There are 3 possibilities, where $(v : c_1^s \cap \dots \cap c_m^s)$ is of the form:

- $(\lambda x : T''_1 \cap \dots \cap T''_n . e'') : \emptyset T''_1 \rightarrow T''^{cl_1} \cap \dots \cap \emptyset T''_m \rightarrow T''^{cl_m}$ such that $T''_1 \cap \dots \cap T''_n \sqsubseteq T'_1 \cap \dots \cap T'_n$ and $e'' \sqsubseteq e'$. Then by rule E-RemoveEmpty and E-AppAbs, $(\lambda x : T''_1 \cap \dots \cap T''_n . e'' : \emptyset T''_1^{cl_1} \cap \dots \cap \emptyset T''_m^{cl_m}) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2} \rightarrow_{\cap CC} ([x \mapsto v_2 : c_{11} \cap \dots \cap c_{m1}] e'') : c_{12} \cap \dots \cap c_{m2}$. As $v_2 \sqsubseteq v'$ and $\vdash_{\cap IC} c_{11} : T''_1$ and ... and $\vdash_{\cap IC} c_{m1} : T''_m$ and $T''_1 \cap \dots \cap T''_m \sqsubseteq T'_1 \cap \dots \cap T'_m$ then $v_2 : c_{11} \cap \dots \cap c_{m1} \sqsubseteq v'$. As $(\lambda x : T'_1 \cap \dots \cap T'_n . e') v' : T'$ and $(\lambda x : T''_1 \cap \dots \cap T''_n . e'') v' \rightarrow_{\cap CC} [x \mapsto v_2] e'$, then by Lemma 4, $\Gamma \vdash_{\cap CC} [x \mapsto v_2] e' : T'$. As $\vdash_{\cap IC} c_{12} : T$ and ... and $\vdash_{\cap IC} c_{m2} : T$ and $T \sqsubseteq T'$ and $e'' \sqsubseteq e'$ then $([x \mapsto v_2 : c_{11} \cap \dots \cap c_{m1}] e'') : c_{12} \cap \dots \cap c_{m2} \sqsubseteq [x \mapsto v_2] e'$.

□

Lemma 10 (Simulation of Unwrapping). *Suppose $\Gamma \vdash_{\cap CC} e_1 : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}$ and $\Gamma \vdash_{\cap CC} e_2 : T_{11} \cap \dots \cap T_{n1}$ and $\Gamma \vdash_{\cap CC} v_1 : cv_1 \cap \dots \cap cv_n : T'_{11} \rightarrow T'_{12} \cap \dots \cap T'_{n1} \rightarrow T'_{n2}$ and $\Gamma \vdash_{\cap CC} v_2 : T'_{11} \cap \dots \cap T'_{n1}$ and $T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2} \sqsubseteq T'_{11} \rightarrow T'_{12} \cap \dots \cap T'_{n1} \rightarrow T'_{n2}$. If $e_1 \sqsubseteq v_1 : cv_1 \cap \dots \cap cv_n$ and $e_2 \sqsubseteq v_2$, then $e_1 e_2 \rightarrow_{\cap CC}^* e$ and $e \sqsubseteq (v_1 : c_1^s \cap \dots \cap c_m^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2}$, with $((c_{11}, c_{12}, c_1^s), \dots, (c_{m1}, c_{m2}, c_m^s)) = \text{simulateArrow}(cv_1, \dots, cv_n)$.*

Theorem 8 (Gradual Guarantee). *Depends on Theorem 6 and Lemmas 2, 3, 4, 6, 9 and 10. If $\Gamma \vdash_{\cap CC} e_1 : T_1$ and $\Gamma \vdash_{\cap CC} e_2 : T_2$ and $e_1 \sqsubseteq e_2$ then:*

1. if $e_2 \rightarrow_{\cap CC} e'_2$ then $e_1 \rightarrow_{\cap CC}^* e'_1$ and $e'_1 \sqsubseteq e'_2$.
2. if $e_1 \rightarrow_{\cap CC} e'_1$ then either $e_2 \rightarrow_{\cap CC}^* e'_2$ and $e'_1 \sqsubseteq e'_2$ or $e_2 \rightarrow_{\cap CC}^* \text{blame}_{T_2} l$.

Proof. (1) We proceed by induction on the length of the derivation tree of $e \sqsubseteq e$.

Base cases:

- Rule for $x \sqsubseteq x$. As $x \rightarrow_{\cap CC} x$ and $x \sqsubseteq x$, it is proved.
- Rule for $n \sqsubseteq n$. As $n \rightarrow_{\cap CC} n$ and $n \sqsubseteq n$, it is proved.
- Rule for $\text{true} \sqsubseteq \text{true}$. As $\text{true} \rightarrow_{\cap CC} \text{true}$ and $\text{true} \sqsubseteq \text{true}$, it is proved.
- Rule for $\text{false} \sqsubseteq \text{false}$. As $\text{false} \rightarrow_{\cap CC} \text{false}$ and $\text{false} \sqsubseteq \text{false}$, it is proved.
- Rule for $e \sqsubseteq \text{blame}_{T'} l$. If $e \sqsubseteq \text{blame}_{T'} l$ then $\Gamma \vdash_{\cap CC} e : T$ and $T \sqsubseteq T'$. If $\text{blame}_{T'} l \rightarrow_{\cap CC} \text{blame}_{T'} l$ then $e \rightarrow_{\cap CC}^* e'$. By Theorem 6, $\Gamma \vdash_{\cap CC} e' : T$, therefore $e' \sqsubseteq \text{blame}_{T'} l$.

Induction step:

- Rule for $\lambda x : T . e \sqsubseteq \lambda x : T' . e'$. If $\lambda x : T . e \sqsubseteq \lambda x : T' . e'$, then $T \sqsubseteq T'$ and $e \sqsubseteq e'$. As $\lambda x : T . e \rightarrow_{\cap CC} \lambda x : T . e$ and $\lambda x : T' . e' \rightarrow_{\cap CC} \lambda x : T' . e'$ and $\lambda x : T . e \sqsubseteq \lambda x : T' . e'$, it is proved.

- Rule for $e_1 \ e_2 \sqsubseteq e'_1 \ e'_2$. There are 6 possibilities:

- Rule E-PushBlame1. If $\Gamma \vdash_{\text{NCC}} e_1 \ e_2 : T$ and $\Gamma \vdash_{\text{NCC}} (\text{blame}_{T'_1 \cap \dots \cap T'_n \rightarrow T'} l) \ e'_2 : T'$ and $e_1 \ e_2 \sqsubseteq (\text{blame}_{T'_1 \cap \dots \cap T'_n \rightarrow T'} l) \ e'_2$ then $\Gamma \vdash_{\text{NCC}} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\text{NCC}} e_2 : T_1 \cap \dots \cap T_n$ and $\Gamma \vdash_{\text{NCC}} \text{blame}_{T'_1 \cap \dots \cap T'_n \rightarrow T'} l : T'_1 \cap \dots \cap T'_n \rightarrow T'$ and $\Gamma \vdash_{\text{NCC}} e'_2 : T'_1 \cap \dots \cap T'_n$ and $e_1 \sqsubseteq \text{blame}_{T'_1 \cap \dots \cap T'_n \rightarrow T'} l$ and $e_2 \sqsubseteq e'_2$. Therefore, $T_1 \cap \dots \cap T_n \rightarrow T \sqsubseteq T'_1 \cap \dots \cap T'_n \rightarrow T'$ and $T \sqsubseteq T'$. If $(\text{blame}_{T'_1 \cap \dots \cap T'_n \rightarrow T'} l) \ e'_2 \rightarrow_{\text{NCC}} \text{blame}_{T'} l$ then $e_1 \ e_2 \rightarrow_{\text{NCC}}^* e$. By Theorem 6, $\Gamma \vdash_{\text{NCC}} e : T$. Therefore, $e \sqsubseteq \text{blame}_{T'} l$.
- Rule E-PushBlame2. If $\Gamma \vdash_{\text{NCC}} e_1 \ e_2 : T$ and $\Gamma \vdash_{\text{NCC}} e'_1 (\text{blame}_{T'_1 \cap \dots \cap T'_n} l) : T'$ and $e_1 \ e_2 \sqsubseteq e'_1 (\text{blame}_{T'_1 \cap \dots \cap T'_n} l)$ then $\Gamma \vdash_{\text{NCC}} e_1 : T_1 \cap \dots \cap T_n \rightarrow T$ and $\Gamma \vdash_{\text{NCC}} e_2 : T_1 \cap \dots \cap T_n$ and $\Gamma \vdash_{\text{NCC}} e'_1 : T'_1 \cap \dots \cap T'_n \rightarrow T'$ and $\Gamma \vdash_{\text{NCC}} \text{blame}_{T'_1 \cap \dots \cap T'_n} l : T'_1 \cap \dots \cap T'_n$ and $e_1 \sqsubseteq e'_1$ and $e_2 \sqsubseteq \text{blame}_{T'_1 \cap \dots \cap T'_n} l$. By Theorem Monotonicity w.r.t. precision of \vdash_{NCC} , as $e_1 \sqsubseteq e'_1$ then $T_1 \cap \dots \cap T_n \rightarrow T \sqsubseteq T'_1 \cap \dots \cap T'_n \rightarrow T'$ and therefore $T \sqsubseteq T'$. If $e'_1 (\text{blame}_{T'_1 \cap \dots \cap T'_n} l) \rightarrow_{\text{NCC}} \text{blame}_{T'} l$ then $e_1 \ e_2 \rightarrow_{\text{NCC}}^* e$. By Theorem 6, $\Gamma \vdash_{\text{NCC}} e : T$. Therefore, $e \sqsubseteq \text{blame}_{T'} l$.
- Rule E-App1. If $e_{11} \ e_{12} \sqsubseteq e_{21} \ e_{22}$ and $e_{21} \ e_{22} \rightarrow_{\text{NCC}} e'_{21} \ e'_{22}$ then $e_{11} \sqsubseteq e_{21}$ and $e_{12} \sqsubseteq e_{22}$ and $e_{21} \rightarrow_{\text{NCC}} e'_{21}$. By the induction hypothesis, $e_{11} \rightarrow_{\text{NCC}} e'_{11}$ and $e'_{11} \sqsubseteq e'_{21}$. Therefore $e_{11} \ e_{12} \rightarrow_{\text{NCC}} e'_{11} \ e'_{12}$. As $e'_{11} \ e'_{12} \sqsubseteq e'_{21} \ e'_{22}$, it is proved.
- Rule E-App2. If $v_{11} \ e_{12} \sqsubseteq v_{21} \ e_{22}$ and $v_{21} \ e_{22} \rightarrow_{\text{NCC}} v'_{21} \ e'_{22}$ then $v_{11} \sqsubseteq v_{21}$ and $e_{12} \sqsubseteq e_{22}$ and $e_{22} \rightarrow_{\text{NCC}} e'_{22}$. By the induction hypothesis, $e_{12} \rightarrow_{\text{NCC}} e'_{12}$ and $e'_{12} \sqsubseteq e'_{22}$. Therefore $v_{11} \ e_{12} \rightarrow_{\text{NCC}} v'_{11} \ e'_{12}$. As $v'_{11} \ e'_{12} \sqsubseteq v'_{21} \ e'_{22}$, it is proved.
- Rule E-AppAbs. If $v_{11} \ v_{12} \sqsubseteq (\lambda x : T_1 \cap \dots \cap T_n . e_2) \ v_2$ and $(\lambda x : T_1 \cap \dots \cap T_n . e_2) \ v_2 \rightarrow_{\text{NCC}} [x \mapsto v_2]e_2$ then $v_{11} \sqsubseteq (\lambda x : T_1 \cap \dots \cap T_n . e_2)$ and $v_{12} \sqsubseteq v_2$. By Lemma 9, $v_{11} \ v_{12} \rightarrow_{\text{NCC}}^* e$, such that $e \sqsubseteq [x \mapsto v_2]e_2$.
- Rule E-SimulateArrow. If $\Gamma \vdash_{\text{NCC}} e_1 \ e_2 : T_{12} \cap \dots \cap T_{n2}$ and $\Gamma \vdash_{\text{NCC}} (v_1 : c_1 \cap \dots \cap c_n) \ v_2 : T'_{12} \cap \dots \cap T'_{n2}$ and $e_1 \ e_2 \sqsubseteq (v_1 : c_1 \cap \dots \cap c_n) \ v_2$ then $\Gamma \vdash_{\text{NCC}} e_1 : T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2}$ and $\Gamma \vdash_{\text{NCC}} e_2 : T_{11} \cap \dots \cap T_{n1}$ and $\Gamma \vdash_{\text{NCC}} v_1 : c_1 \cap \dots \cap c_n : T'_{11} \rightarrow T'_{12} \cap \dots \cap T'_{n1} \rightarrow T'_{n2}$ and $\Gamma \vdash_{\text{NCC}} v_2 : T'_{11} \cap \dots \cap T'_{n1}$ and $e_1 \sqsubseteq v_1 : c_1 \cap \dots \cap c_n$ and $e_2 \sqsubseteq v_2$. Therefore, $\vdash_{\text{NIC}} c_1 : T'_{11} \rightarrow T'_{12}$ and ... and $\vdash_{\text{NIC}} c_n : T'_{n1} \rightarrow T'_{n2}$ and as $e_1 \sqsubseteq v_1 : c_1 \cap \dots \cap c_n$, then $T_{11} \rightarrow T_{12} \cap \dots \cap T_{n1} \rightarrow T_{n2} \sqsubseteq T'_{11} \rightarrow T'_{12} \cap \dots \cap T'_{n1} \rightarrow T'_{n2}$. By Lemma 10, $e_1 \ e_2 \rightarrow_{\text{NCC}}^* e$ and $e \sqsubseteq (v_1 : c_1^s \cap \dots \cap c_n^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2}$. As $(v_1 : c_1 \cap \dots \cap c_n) \ v_2 \rightarrow_{\text{NCC}} (v_1 : c_1^s \cap \dots \cap c_n^s) (v_2 : c_{11} \cap \dots \cap c_{m1}) : c_{12} \cap \dots \cap c_{m2}$, it is proved.

- Rule for $e : c_1 \cap \dots \cap c_n \sqsubseteq e' : c'_1 \cap \dots \cap c'_n$. There are 5 possibilities:

- Rule E-PushBlameCast. If $\Gamma \vdash_{\text{NCC}} e : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n$ and $\Gamma \vdash_{\text{NCC}} \text{blame}_{T'} l : c'_1 \cap \dots \cap c'_n : T'_1 \cap \dots \cap T'_n$ and $e : c_1 \cap \dots \cap c_n \sqsubseteq \text{blame}_{T'} l : c'_1 \cap \dots \cap c'_n$ then $e \sqsubseteq \text{blame}_{T'} l$ and $\vdash_{\text{NIC}} c_1 : T_1$ and ... and $\vdash_{\text{NIC}} c_n : T_n$ and $\vdash_{\text{NIC}} c'_1 : T'_1$ and ... and $\vdash_{\text{NIC}} c'_n : T'_n$ and $c_1 \sqsubseteq c'_1$ and ... and $c_n \sqsubseteq c'_n$. By Lemma Monotonicity w.r.t precision of IC, $T_1 \sqsubseteq T'_1$ and ... and $T_n \sqsubseteq T'_n$. If $\text{blame}_{T'} l : c'_1 \cap \dots \cap c'_n \rightarrow_{\text{NCC}} \text{blame}_{T'_1 \cap \dots \cap T'_n} l$ then $e : c_1 \cap \dots \cap c_n \rightarrow_{\text{NCC}}^* e'$ By Theorem 6, $\Gamma \vdash_{\text{NCC}} e' : T_1 \cap \dots \cap T_n$. As $\Gamma \vdash_{\text{NCC}} \text{blame}_{T'_1 \cap \dots \cap T'_n} l : T'_1 \cap \dots \cap T'_n$ then $e' \sqsubseteq \text{blame}_{T'_1 \cap \dots \cap T'_n} l$.
- Rule E-Evaluate. If $e : c_1 \cap \dots \cap c_n \sqsubseteq e' : c'_1 \cap \dots \cap c'_n$ then $e \sqsubseteq e'$ and $c_1 \sqsubseteq c'_1$ and ... and $c_n \sqsubseteq c'_n$. If $e' : c'_1 \cap \dots \cap c'_n \rightarrow_{\text{NCC}} e'_1 : c'_1 \cap \dots \cap c'_n$, then $e' \rightarrow_{\text{NCC}} e'_1$. By the induction hypothesis, $e \rightarrow_{\text{NCC}} e_1$ and $e_1 \sqsubseteq e'_1$. Therefore, by rule E-Evaluate, $e : c_1 \cap \dots \cap c_n \rightarrow_{\text{NCC}} e_1 : c_1 \cap \dots \cap c_n$. Then $e_1 : c_1 \cap \dots \cap c_n \sqsubseteq e'_1 : c'_1 \cap \dots \cap c'_n$.

- Rule E-EvaluateCasts. If $v : c_1 \cap \dots \cap c_n \sqsubseteq v' : c'_1 \cap \dots \cap c'_n$, then $v \sqsubseteq v'$ and $c_1 \sqsubseteq c'_1$ and ... and $c_n \sqsubseteq c'_n$. If $v' : c'_1 \cap \dots \cap c'_n \rightarrow_{\text{OCC}} v' : cv'_1 \cap \dots \cap cv'_n$ then $c'_1 \rightarrow_{\text{OIC}} cv'_1$ and ... and $c'_n \rightarrow_{\text{OIC}} cv'_n$. By Lemma 6, $c_1 \rightarrow_{\text{OIC}} cv_1$ and $cv_1 \sqsubseteq cv'_1$ and ... and $c_n \rightarrow_{\text{OIC}} cv_n$ and $cv_n \sqsubseteq cv'_n$. Therefore, $v : c_1 \cap \dots \cap c_n \rightarrow_{\text{OCC}} v : cv_1 \cap \dots \cap cv_n$. As $v \sqsubseteq v'$ and $cv_1 \sqsubseteq cv'_1$ and ... and $cv_n \sqsubseteq cv'_n$, then $v : cv_1 \cap \dots \cap cv_n \sqsubseteq v' : cv'_1 \cap \dots \cap cv'_n$.
- Rule E-PropagateBlame. If $v : \text{blame } I_1 F_1 l_1^{cl_1} \cap \dots \cap \text{blame } I_n F_n l_n^{cl_n} \sqsubseteq v' : \text{blame } I'_1 F'_1 l'_1^{cl'_1} \cap \dots \cap \text{blame } I'_n F'_n l'_n^{cl'_n}$ then $v \sqsubseteq v'$ and $\text{blame } I_1 F_1 l_1^{cl_1} \sqsubseteq \text{blame } I'_1 F'_1 l'_1^{cl'_1}$ and ... and $\text{blame } I_n F_n l_n^{cl_n} \sqsubseteq \text{blame } I'_n F'_n l'_n^{cl'_n}$. Therefore, $F_1 \sqsubseteq F'_1$ and ... and $F_n \sqsubseteq F'_n$. If $v' : \text{blame } I'_1 F'_1 l'_1^{cl'_1} \cap \dots \cap \text{blame } I'_n F'_n l'_n^{cl'_n} \rightarrow_{\text{OCC}} \text{blame}_{F'_1 \cap \dots \cap F'_n} l'_1$, then $v : \text{blame } I_1 F_1 l_1^{cl_1} \cap \dots \cap \text{blame } I_n F_n l_n^{cl_n} \rightarrow_{\text{OCC}} \text{blame}_{F_1 \cap \dots \cap F_n} l_1$. As $F_1 \cap \dots \cap F_n \sqsubseteq F'_1 \cap \dots \cap F'_n$, then $\text{blame}_{F_1 \cap \dots \cap F_n} l_1 \sqsubseteq \text{blame}_{F'_1 \cap \dots \cap F'_n} l'_1$.
- Rule E-RemoveEmpty. If $v : \emptyset T_1^{cl_1} \cap \dots \cap \emptyset T_n^{cl_n} \sqsubseteq v' : \emptyset T'_1^{cl'_1} \cap \dots \cap \emptyset T'_n^{cl'_n}$ then $v \sqsubseteq v'$. If $v' : \emptyset T'_1^{cl'_1} \cap \dots \cap \emptyset T'_n^{cl'_n} \rightarrow_{\text{OCC}} v'$ then $v : \emptyset T_1^{cl_1} \cap \dots \cap \emptyset T_n^{cl_n} \rightarrow_{\text{OCC}} v$. As $v \sqsubseteq v'$, it is proved.

• Rule for $e : c_1 \cap \dots \cap c_n \sqsubseteq e'$.

• Rule for $e \sqsubseteq e' : c'_1 \cap \dots \cap c'_n$. There are 5 possibilities:

- Rule E-PushBlameCast. If $\Gamma \vdash_{\text{OCC}} e : T$, $\Gamma \vdash_{\text{OCC}} \text{blame}_{T'} l : c_1 \cap \dots \cap c_n : T_1 \cap \dots \cap T_n$ and $e \sqsubseteq \text{blame}_{T'} l : c_1 \cap \dots \cap c_n$ then $e \sqsubseteq \text{blame}_{T'} l$ and $T \sqsubseteq T_1 \cap \dots \cap T_n$. If $\text{blame}_{T'} l : c_1 \cap \dots \cap c_n \rightarrow_{\text{OCC}} \text{blame}_{T_1 \cap \dots \cap T_n} l_1$ then $e \rightarrow_{\text{OCC}}^* e'$. By Theorem 6, $\Gamma \vdash_{\text{OCC}} e' : T$. As $\Gamma \vdash_{\text{OCC}} \text{blame}_{T_1 \cap \dots \cap T_n} l_1 : T_1 \cap \dots \cap T_n$, then $e' \sqsubseteq \text{blame}_{T_1 \cap \dots \cap T_n} l_1$.
- Rule E-Evaluate. If $e \sqsubseteq e' : c'_1 \cap \dots \cap c'_n$ then $e \sqsubseteq e'$, $\Gamma \vdash_{\text{OCC}} e : T$, $\vdash_{\text{OIC}} c'_1 : T_1$ and ... and $\vdash_{\text{OIC}} c'_n : T_n$ and $T \sqsubseteq T_1 \cap \dots \cap T_n$. By rule E-Evaluate, $e' \rightarrow_{\text{OCC}} e'_1$, therefore, by the induction hypothesis, $e \rightarrow_{\text{OCC}} e_1$ and $e_1 \sqsubseteq e'_1$. By rule E-Evaluate, $e' : c'_1 \cap \dots \cap c'_n \rightarrow_{\text{OCC}} e'_1 : c'_1 \cap \dots \cap c'_n$. By Lemma 4, $\Gamma \vdash_{\text{OCC}} e_1 : T$, therefore $e_1 \sqsubseteq e'_1 : c'_1 \cap \dots \cap c'_n$.
- Rule E-EvaluateCasts. If $e \sqsubseteq v : c_1 \cap \dots \cap c_n$, then $e \sqsubseteq v$, $\Gamma \vdash_{\text{OCC}} e : T$, $\vdash_{\text{OIC}} c_1 : T_1$ and ... and $\vdash_{\text{OIC}} c_n : T_n$ and $T \sqsubseteq T_1 \cap \dots \cap T_n$. If $v : c_1 \cap \dots \cap c_n \rightarrow_{\text{OCC}} v : cv_1 \cap \dots \cap cv_n$ then, by rule E-EvaluateCasts, $c_1 \rightarrow_{\text{OIC}} cv_1$ and ... and $c_n \rightarrow_{\text{OIC}} cv_n$, and by Lemmas 2 and 3, $\vdash_{\text{OIC}} cv_1 : T_1$ and ... and $\vdash_{\text{OIC}} cv_n : T_n$. If $v : c_1 \cap \dots \cap c_n \rightarrow_{\text{OCC}} v : cv_1 \cap \dots \cap cv_n$ then $e \rightarrow_{\text{OCC}}^* e_1$. By Theorem 6, $\Gamma \vdash_{\text{OCC}} e_1 : T$, therefore $e_1 \sqsubseteq v : cv_1 \cap \dots \cap cv_n$.
- Rule E-PropagateBlame. If $\Gamma \vdash_{\text{OCC}} e : T$, $\Gamma \vdash_{\text{OCC}} v : \text{blame } I_1 F_1 l_1^{cl_1} \cap \dots \cap \text{blame } I_n F_n l_n^{cl_n} : F_1 \cap \dots \cap F_n$ and $e \sqsubseteq v : \text{blame } I_1 F_1 l_1^{cl_1} \cap \dots \cap \text{blame } I_n F_n l_n^{cl_n}$ then $e \sqsubseteq v$ and $T \sqsubseteq F_1 \cap \dots \cap F_n$. If $v : \text{blame } I_1 F_1 l_1^{cl_1} \cap \dots \cap \text{blame } I_n F_n l_n^{cl_n} \rightarrow_{\text{OCC}} \text{blame}_{F_1 \cap \dots \cap F_n} l_1$ then $e \rightarrow_{\text{OCC}}^* e'$. By Theorem 6, $\Gamma \vdash_{\text{OCC}} e' : T$. As $\Gamma \vdash_{\text{OCC}} \text{blame}_{F_1 \cap \dots \cap F_n} l_1 : F_1 \cap \dots \cap F_n$, then $e' \sqsubseteq \text{blame}_{F_1 \cap \dots \cap F_n} l_1$.
- Rule E-RemoveEmpty. If $e \sqsubseteq v : \emptyset T_1^{cl_1} \cap \dots \cap \emptyset T_n^{cl_n}$ then $e \sqsubseteq v$. By Lemma 8, $e \rightarrow_{\text{OCC}}^* e'$ and $e' \sqsubseteq v$. If $v : \emptyset T_1^{cl_1} \cap \dots \cap \emptyset T_n^{cl_n} \rightarrow_{\text{OCC}} v$, then it is proved.

(2) Let's assume that $e_1 \rightarrow_{\text{OCC}} e'_1$. Because e_2 is well typed, then either $e_2 \rightarrow_{\text{OCC}}^* e'_2$ or $e_2 \rightarrow_{\text{OCC}}^* \text{blame}_{T_2} l$. If $e_2 \rightarrow_{\text{OCC}}^* e'_2$ then, by part 1, $e'_1 \sqsubseteq e'_2$. If $e_2 \rightarrow_{\text{OCC}}^* \text{blame}_{T_2} l$, we are done.

□

References

- [1] Mario Coppo, Mariangiola Dezani-Ciancaglini, et al. An extension of the basic functionality theory for the λ -calculus. *Notre Dame journal of formal logic*, 21(4):685–693, 1980.