Supplement to: The key to blame: Gradual typing meets cryptography

Jeremy Siek
Indiana University, USA
jsiek@indiana.edu

Philip Wadler
University of Edinburgh, UK
wadler@inf.ed.ac.uk

1. Examples of λB

Example programs (1)–(7) from Section 2.1 and 2.2 of the paper are worked out in detail in Figures 1, 2, and 3. Readers interested in following the details may wish to begin with Figure 3, which is simpler than the other two. We write $\longrightarrow^n$ to indicate the result of n reduction steps.

2. Type Safety of λB

Lemma 1. *If $\forall X.A \prec B$, then $A[X{:=}\star] \prec B$.*

Proof. We proceed by induction on $\forall X.A \prec B$.

- Case $\boxed{A' \prec \forall Y.B'}$ (where $A' = \forall X.A$):
  We have $\forall X.A \prec B'$. By the induction hypothesis, $A[X{:=}\star] \prec B'$. Therefore, $A[X{:=}\star] \prec \forall Y.B'[Y]$.
- Case $\boxed{\forall X.A \prec B}$:
  We have $A[X{:=}\star] \prec B$, which completes this case.
- Case $\boxed{A' \prec \star}$ (where $A' = \forall X.A$):
  We immediatley have $A[X{:=}\star] \prec \star$.

□

Lemma 2 (Subject Reduction)**.** *If $\Sigma \vdash M : A$ and $M \longmapsto N$, then $\Sigma \vdash N : A$.*

Proof. The proof is by cases on $M \longmapsto N$. Many of the cases are trivial or standard. We give the cases that are novel or non-trivial.

- Case $\boxed{(\Lambda X.\, V)\; X \longmapsto V}$:
  We immediately have $\Sigma \vdash V : A$.
- Case $\boxed{\begin{array}{ll} & (V : A' \to B \overset{\phi}{\Longrightarrow} C \to D)\; W \\ \longmapsto & V\;(W : C \overset{-\phi}{\Longrightarrow} A') : B \overset{\phi}{\Longrightarrow} D \end{array}}$:
  We have $A'{\to}B \prec^{\phi} C{\to}D$. So $C \prec^{-\phi} A'$ and $B \prec^{\phi} D$. Thus, the RHS also has type $D = A$.
- Case $\boxed{\begin{array}{ll} & (V : \forall X.A' \overset{\phi}{\Longrightarrow} \forall X.B)\; X \\ \longmapsto & V\; X : A' \overset{\phi}{\Longrightarrow} B \end{array}}$:
  We have $\forall X.A' \prec^{\phi} \forall X.B$, so $A' \prec^{\phi} B$. Also, we have $\Sigma \vdash V\; X : A'$, so the RHS has type $B = A$.
- Case $\boxed{(V : A' \overset{p}{\Longrightarrow} \forall X.B)\; X \longmapsto V : A' \overset{p}{\Longrightarrow} B}$:
  We have $A' \prec \forall X.B$, so $A' \prec B$. Thus, the RHS has type $B = A$.
- Case $\boxed{V : \forall X.A' \overset{p}{\Longrightarrow} B \longmapsto (V\; \star) : A'[X{:=}\star] \overset{p}{\Longrightarrow} B}$:
  We have $\forall X.A' \prec B$, so $A'[X{:=}\star] \prec B$ by Lemma 1. Thus, the RHS has type B.

□

Definition 3. *Well-typed contexts, written $\Sigma \rhd \mathcal{E} : B \Rightarrow A$, are defined in the usual way.*

Lemma 4 (Decomposition)**.** *If $\Sigma \vdash M : A$, then either*

1. *$M = V'$,*
2. *$M = \mathcal{E}'[\texttt{blame}\, p']$,*
3. *$M = \mathcal{E}'[\nu\alpha{:=}A'.N']$, or*
4. *$M = \mathcal{E}'[M']$ and $M' \longmapsto N'$.*

where all primed variables are existentially quantified.

Proof. The proof is by induction on $M : A$.

- $\boxed{\dfrac{}{\Gamma \vdash c : \mathtt{type}(c)}}$ Pick V' to be c.
- $\boxed{\dfrac{op : \vec{A} \to B \quad \Gamma \vdash \vec{M} : \vec{A}}{\Gamma \vdash op(\vec{M}) : B}}$
  If $\vec{M}$ are all values, then we have $op(\vec{V}) \longmapsto [\![op]\!](\vec{V})$ (We require the primitive operators to be type safe.) Pick $\mathcal{E}' = \square$, $M' = op(\vec{V})$, and $N' = [\![op]\!](\vec{V})$ to conclude.
  If one of $\vec{M}$ is not a value, let M_i be the first such. Then either $M_i = \mathcal{E}'_1[\texttt{blame}\, p']$ or $M_i = \mathcal{E}'_1[M'_i]$ and $M'_i \longmapsto N'_i$. Either way, we pick $\mathcal{E}' = op(\vec{V}, \mathcal{E}'_1, \vec{M})$ In the first case we have $\Sigma \rhd \mathcal{E}'[\texttt{blame}\, p'] \longmapsto \Sigma \vdash \texttt{blame}\, p'$. In the second case we have $\Sigma \rhd \mathcal{E}'[M'_i] \longmapsto \Sigma \vdash \mathcal{E}'[N'_i]$.
- $\boxed{\dfrac{A : \mathtt{tp} \quad \Gamma, x : A \vdash N : B}{\Gamma \vdash (\lambda x{:}A.\, N) : A \to B}}$ Pick V' to be $(\lambda x{:}A.\, N[x])$.
- $\boxed{\dfrac{\Gamma, X : \mathtt{tp} \vdash V : B}{\Gamma \vdash (\Lambda X.\, V) : \forall X.B}}$ Pick V' to be $(\Lambda X.\, V)$.
- $\boxed{\dfrac{A : \mathtt{tp} \;\; \Gamma, X{:=}A \vdash N : B \;\; X \notin B}{\Gamma \vdash (\nu X{:=}A.N) : B}}$
  We satisfy the third option by picking $\mathcal{E}' = \square$.
- $\boxed{\dfrac{\Gamma \vdash L : A \to B \qquad \Gamma \vdash M_1 : A}{\Gamma \vdash (L\; M_1) : B}}$
  - If L and M_1 are values, then pick $\mathcal{E}' = \square$ and $M' = (L\; M_1)$. By canonical forms, L is in one of the following forms:
    1. $L = \lambda x{:}A.\, N_1$,
    2. $L = V : A' \to B' \overset{\phi}{\Longrightarrow} A \to B$, or
    3. $L = V : A' \to B' \overset{p}{\Longrightarrow} A \to B$.

    In each of these cases, a reduction rule applies, so we have $M' \longrightarrow N'$ for some N'.

Example 1

$$\begin{aligned} \mathit{two} &= (\Lambda X.\, \lambda f{:}X{\to}X.\, \lambda x{:}X.\, f\ (f\ x)) & \mathit{inc}^\star &= \lceil \lambda x.\, x+1 \rceil \\ \mathit{two}_X &= (\lambda f{:}X{\to}X.\, \lambda x{:}X.\, f\ (f\ x)) & \mathit{inc}^{\star\to\star} &= (\lambda x{:}\star.\, \lceil x+1 \rceil) \\ \mathit{two}^\star &= (\mathit{two} : \forall X.(X{\to}X){\to}X{\to}X \xRightarrow{+\ell} \star) \end{aligned}$$

$$\begin{array}{ll}
 & \bullet \rhd \lceil \mathit{two}^\star\ \mathit{inc}^\star\ 0 \rceil \\
\overset{\text{def}}{=} & \bullet \rhd (((\mathit{two} : \forall X.(X{\to}X){\to}X{\to}X \xRightarrow{+\ell} \star \xRightarrow{+m} \star{\to}\star)\ \mathit{inc}^\star) : \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & \bullet \rhd (((\mathit{two}\ \star : (\star{\to}\star){\to}\star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+m} \star{\to}\star)\ \mathit{inc}^\star) : \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\overset{\text{def}}{=} & \bullet \rhd ((((\nu X{:=}\star.\mathit{two}\ X : (X{\to}X){\to}X{\to}X \xRightarrow{+X} (\star{\to}\star){\to}\star{\to}\star) : (\star{\to}\star){\to}\star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+m} \star{\to}\star)\ \mathit{inc}^\star) : \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\mathit{two}\ X : (X{\to}X){\to}X{\to}X \xRightarrow{+X} (\star{\to}\star){\to}\star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+m} \star{\to}\star)\ \mathit{inc}^\star) : \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\mathit{two}_X : (X{\to}X){\to}X{\to}X \xRightarrow{+X} (\star{\to}\star){\to}\star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+m} \star{\to}\star)\ \mathit{inc}^\star) : \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\mathit{two}_X : (X{\to}X){\to}X{\to}X \xRightarrow{+X} (\star{\to}\star){\to}\star{\to}\star \xRightarrow{+\ell} \star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+m} \star{\to}\star)\ \mathit{inc}^\star) : \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\mathit{two}_X : (X{\to}X){\to}X{\to}X \xRightarrow{+X} (\star{\to}\star){\to}\star{\to}\star \xRightarrow{+\ell} \star{\to}\star)\ \mathit{inc}^\star) : \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\mathit{two}_X : (X{\to}X){\to}X{\to}X \xRightarrow{+X} (\star{\to}\star){\to}\star{\to}\star)\ (\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star)) : \star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd ((\mathit{two}_X\ (\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X)) : X{\to}X \xRightarrow{+X} \star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\lambda x{:}X.\, (\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ x) : X{\to}X \xRightarrow{+X} \star{\to}\star \xRightarrow{+\ell} \star \xRightarrow{+n} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\lambda x{:}X.\, (\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ x) : X{\to}X \xRightarrow{+X} \star{\to}\star)\ \lceil 0 \rceil \\
\longrightarrow & X{:=}\star \rhd (((\lambda x{:}X.\, (\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ x)\ (\lceil 0 \rceil : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd (((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\lceil 0 \rceil : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\overset{\text{def}}{=} & X{:=}\star \rhd (((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^{\star\to\star} : \star{\to}\star \xRightarrow{+o} \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\lceil 0 \rceil : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd (((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^{\star\to\star} : \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\lceil 0 \rceil : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd (((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^{\star\to\star}\ (\lceil 0 \rceil : \star \xRightarrow{-X} X \xRightarrow{+X} \star)) : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd (((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\mathit{inc}^{\star\to\star}\ \lceil 0 \rceil) : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow^3 & X{:=}\star \rhd (((\mathit{inc}^\star : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\lceil 1 \rceil : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\overset{\text{def}}{=} & X{:=}\star \rhd (((\mathit{inc}^{\star\to\star} : \star{\to}\star \xRightarrow{+o} \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\lceil 1 \rceil : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd (((\mathit{inc}^{\star\to\star} : \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\lceil 1 \rceil : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd (\mathit{inc}^{\star\to\star}\ (\lceil 1 \rceil : \star \xRightarrow{-X} X \xRightarrow{+X} \star)) :: \star \xRightarrow{-X} X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd (\mathit{inc}^{\star\to\star}\ \lceil 1 \rceil) : \star \xRightarrow{-X} X \xRightarrow{+X} \star \\
\longrightarrow^3 & X{:=}\star \rhd \lceil 2 \rceil : \star \xRightarrow{-X} X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd \lceil 2 \rceil
\end{array}$$

Example 2

$$\begin{array}{ll}
 & \bullet \rhd \lceil \mathit{two}^\star\ 0\ \mathit{inc}^\star \rceil \\
\longrightarrow^{11} & X{:=}\star \rhd (((\lceil 0 \rceil : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((\lceil 0 \rceil : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\mathit{inc}^\star : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\overset{\text{def}}{=} & X{:=}\star \rhd (((\lceil 0 \rceil : \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ ((0 : \mathtt{num} \xRightarrow{+o} \star \xRightarrow{-\ell} \star{\to}\star \xRightarrow{-X} X{\to}X))\ (\mathit{inc}^\star : \star \xRightarrow{-X} X)) : X \xRightarrow{+X} \star \\
\longrightarrow & X{:=}\star \rhd \mathtt{blame}\ -\ell
\end{array}$$

Figure 1. Example, untyped code with typed component

- If L is a value but not M_1, then we apply the induction hypothesis for $M_1 : A$ to obtain a decomposition $\mathcal{E}''$ of M_1 and then pick $\mathcal{E}' = (L\ \mathcal{E}'')$.
- If L is not a value, then we apply the induction hypothesis for $L : A \to B$ to obtain a decomposition $\mathcal{E}''$ of L and then pick $\mathcal{E}' = (\mathcal{E}''\ M_1)$.

- $$\boxed{\dfrac{\Gamma \vdash L : \forall X.B \qquad X{:=}A \in \Gamma}{\Gamma \vdash (L\ X) : B}}$$
  - If L is a value, then pick $\mathcal{E}' = \square$ and $M' = (L\ \alpha)$. By canonical forms, L is in one of the following forms:
    1. $L = (\Lambda X.\, V')$,
    2. $L = V' : \forall X.A \xRightarrow{\phi} \forall X.B$, or
    3. $L = V' : A \xRightarrow{p} \forall X.B$.

    In each of these cases, a reduction rule applies, so we have $M' \longrightarrow N'$ for some N'.
  - If L is not a value, we apply the induction hypothesis to obtain a decomposition $\mathcal{E}''$ of L and then pick $\mathcal{E}' = (\mathcal{E}''\ \alpha)$.

- $$\boxed{\dfrac{\Gamma \vdash M_1 : A \qquad \Gamma \vdash A \prec^\phi B}{\Gamma \vdash (M_1 : A \xRightarrow{\phi} B) : B}}$$
  - If M_1 is a value V, we proceed by cases on $A \prec^\phi B$.
    1. Case $\iota \prec^\phi \iota$:
       Pick $\mathcal{E}' = \square$ and $M' = (V : \iota \xRightarrow{\phi} \iota)$.
       $$V : \iota \xRightarrow{\phi} \iota \longmapsto V$$
    2. Case $A_1{\to}A_2 \prec^\phi B_1{\to}B_2$:
       $(V : A_1{\to}A_2 \xRightarrow{\phi} B_1{\to}B_2)$ is a value.

Example 3

$$\mathit{two}^\star = \lceil \lambda f.\,\lambda x.\,f\;(f\;x)\rceil \qquad \mathit{two} = (\mathit{two}^\star : \star \xRightarrow{+\ell} \forall X.(X\to X)\to X\to X)$$
$$\mathit{two}^{\star\to\star} = (\lambda f{:}\,\star.\,\lceil\lambda x.\,f\;(f\;x)\rceil) \qquad \mathit{inc} = (\lambda x{:}\texttt{num}.\,x+1)$$

$$\begin{array}{ll}
 & \bullet \triangleright \mathit{two}\;\texttt{num}\;\mathit{inc}\;0 \\
\overset{\text{def}}{=} & \bullet \triangleright (\mathit{two}^\star : \star \xRightarrow{+\ell} \forall X.(X\to X)\to X\to X)\;\texttt{num}\;\mathit{inc}\;0 \\
\overset{\text{def}}{=} & \bullet \triangleright (\nu X{:=}\texttt{num}.(\mathit{two}^\star : \star \xRightarrow{+\ell} \forall X.(X\to X)\to X\to X)\;X : (X\to X)\to X\to X \xRightarrow{+X} (\texttt{num}\to\texttt{num})\to\texttt{num}\to\texttt{num})\;\mathit{inc}\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright (\mathit{two}^\star : \star \xRightarrow{+\ell} (X\to X)\to X\to X \xRightarrow{+X} (\texttt{num}\to\texttt{num})\to\texttt{num}\to\texttt{num})\;\mathit{inc}\;0 \\
\overset{\text{def}}{=} & X{:=}\texttt{num} \triangleright (\mathit{two}^{\star\to\star} : \star\to\star \xRightarrow{+m} \star \xRightarrow{+\ell} (X\to X)\to X\to X \xRightarrow{+X} (\texttt{num}\to\texttt{num})\to\texttt{num}\to\texttt{num})\;\mathit{inc}\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright (\mathit{two}^{\star\to\star} : \star\to\star \xRightarrow{+m} \star \xRightarrow{+\ell} \star\to\star \xRightarrow{+\ell} (X\to X)\to X\to X \xRightarrow{+X} (\texttt{num}\to\texttt{num})\to\texttt{num}\to\texttt{num})\;\mathit{inc}\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright (\mathit{two}^{\star\to\star} : \star\to\star \xRightarrow{+\ell} (X\to X)\to X\to X \xRightarrow{+X} (\texttt{num}\to\texttt{num})\to\texttt{num}\to\texttt{num})\;\mathit{inc}\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright (((\mathit{two}^{\star\to\star} : \star\to\star \xRightarrow{+\ell} (X\to X)\to X\to X)\;(\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X)) : X\to X \xRightarrow{+X} \texttt{num}\to\texttt{num})\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright (\mathit{two}^{\star\to\star}\;(\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star) : \star \xRightarrow{+\ell} X\to X \xRightarrow{+X} \texttt{num}\to\texttt{num})\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright (\mathit{two}^{\star\to\star}\;(\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star) : \star \xRightarrow{+\ell} X\to X \xRightarrow{+X} \texttt{num}\to\texttt{num})\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright (\mathit{two}^{\star\to\star}\;(\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star) : \star \xRightarrow{+\ell} X\to X \xRightarrow{+X} \texttt{num}\to\texttt{num})\;0 \\
\longrightarrow & X{:=}\texttt{num} \triangleright ((\lambda x{:}\,\star\,.\,(\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+n} \star\to\star) \\
 & \qquad ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+o} \star\to\star)\;x)) : \star\to\star \xRightarrow{+p} \star \xRightarrow{+\ell} X\to X \xRightarrow{+X} \texttt{num}\to\texttt{num})\;0 \\
\longrightarrow^2 & X{:=}\texttt{num} \triangleright ((\lambda x{:}\,\star\,.\,(\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+n} \star\to\star) \\
 & \qquad ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+o} \star\to\star)\;x)) : \star\to\star \xRightarrow{+\ell} X\to X \xRightarrow{+X} \texttt{num}\to\texttt{num})\;0 \\
\longrightarrow^2 & X{:=}\texttt{num} \triangleright ((\lambda x{:}\,\star\,.\,(\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+n} \star\to\star) \\
 & \qquad ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+o} \star\to\star)\;x))\;(0 : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star)) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow & X{:=}\texttt{num} \triangleright ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+n} \star\to\star) \\
 & \qquad ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star \xRightarrow{-\ell} \star \xRightarrow{+o} \star\to\star)\;(0 : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star))) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow^4 & X{:=}\texttt{num} \triangleright ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star) \\
 & \qquad ((\mathit{inc}\;(0 : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num})) : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star)) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow^2 & X{:=}\texttt{num} \triangleright ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star)\;((\mathit{inc}\;0) : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star)) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow & X{:=}\texttt{num} \triangleright ((\mathit{inc} : \texttt{num}\to\texttt{num} \xRightarrow{-X} X\to X \xRightarrow{-\ell} \star\to\star)(1 : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star)) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow^2 & X{:=}\texttt{num} \triangleright (\mathit{inc}\;(1 : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num})) : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow^2 & X{:=}\texttt{num} \triangleright (\mathit{inc}\;1) : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow & X{:=}\texttt{num} \triangleright 2 : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow^2 & X{:=}\texttt{num} \triangleright 2
\end{array}$$

Example 4

$$\begin{array}{ll}
 & \bullet \triangleright (\lceil \lambda f.\,\lambda x.\,2\rceil : \star \xRightarrow{+\ell} \forall X.(X\to X)\to X\to X)\;\texttt{num}\;\mathit{inc}\;0 \\
\longrightarrow^{10} & X{:=}\texttt{num} \triangleright ((\lambda x{:}\,\star\,.\,\lceil 2\rceil) : \star\to\star \xRightarrow{+\ell} X\to X \xRightarrow{+X} \texttt{num}\to\texttt{num})\;0 \\
\longrightarrow^2 & X{:=}\texttt{num} \triangleright ((\lambda x{:}\,\star\,.\,\lceil 2\rceil)\;(0 : \texttt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow & X{:=}\texttt{num} \triangleright \lceil 2\rceil : \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\overset{\text{def}}{=} & X{:=}\texttt{num} \triangleright 2 : \texttt{num} \xRightarrow{+m} \star \xRightarrow{+\ell} X \xRightarrow{+X} \texttt{num} \\
\longrightarrow & X{:=}\texttt{num} \triangleright \texttt{blame}\;+\ell
\end{array}$$

Figure 2. Example, typed code with untyped component

3. Case $\forall X.A \prec^\phi \forall X.B$:
   $(V : \forall X.A \xRightarrow{\phi} \forall X.B)$ is a value.
4. Case $X \prec^{+X} B$:
   So $V : X$ and by canonical forms, $V = V' : B \xRightarrow{-X} X$.
   Pick $\mathcal{E}' = \Box$ and $M' = (V' : B \xRightarrow{-X} X \xRightarrow{+X} B)$.
   $$(V' : B \xRightarrow{-X} X \xRightarrow{+X} B) \longmapsto V'$$
5. Case $A \prec^{-X} X$:
   $(V : A \xRightarrow{-X} X)$ is a value.
6. Case $X \prec^\phi X$:
   Pick $\mathcal{E}' = \Box$ and $M' = (V : X \xRightarrow{\phi} X)$.
   $$(V : X \xRightarrow{\phi} X) \longmapsto V$$
7. Case $\star \prec^\phi \star$:
   Pick $\mathcal{E}' = \Box$ and $M' = (V : \star \xRightarrow{\phi} \star)$.
   $$(V : \star \xRightarrow{\phi} \star) \longmapsto V$$

Example 5

$$\begin{array}{ll}
 & \bullet \rhd (\lceil \lambda x.\, x \rceil : \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\stackrel{\text{def}}{=} & \bullet \rhd ((\lambda x{:}\, \star\, .\, x) : \star \to \star \xRightarrow{+o} \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\longrightarrow & \bullet \rhd ((\lambda x{:}\, \star\, .\, x) : \star \to \star \xRightarrow{+o} \star \xRightarrow{+\ell} \star \to \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\longrightarrow & \bullet \rhd ((\lambda x{:}\, \star\, .\, x) : \star \to \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\stackrel{\text{def}}{=} & \bullet \rhd (\nu X{:=}\mathtt{num}. ((\lambda x{:}\, \star\, .\, x) : \star \to \star \xRightarrow{+\ell} \forall X. X \to X)\ X : X \to X \xRightarrow{+X} \mathtt{num} \to \mathtt{num})\ 1 \\
\longrightarrow & X{:=}\mathtt{num} \rhd (((\lambda x{:}\, \star\, .\, x) : \star \to \star \xRightarrow{+\ell} \forall X. X \to X)\ X : X \to X \xRightarrow{+X} \mathtt{num} \to \mathtt{num})\ 1 \\
\longrightarrow & X{:=}\mathtt{num} \rhd ((\lambda x{:}\, \star\, .\, x) : \star \to \star \xRightarrow{+\ell} X \to X \xRightarrow{+X} \mathtt{num} \to \mathtt{num})\ 1 \\
\longrightarrow & X{:=}\mathtt{num} \rhd (((\lambda x{:}\, \star\, .\, x) : \star \to \star \xRightarrow{+\ell} X \to X)\ (1 : \mathtt{num} \xRightarrow{-X} X)) : X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd ((\lambda x{:}\, \star\, .\, x)\ (1 : \mathtt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star)) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd 1 : \mathtt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star \xRightarrow{+\ell} X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd 1 : \mathtt{num} \xRightarrow{-X} X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd 1
\end{array}$$

Example 6

$$\begin{array}{ll}
 & \bullet \rhd (\lceil \lambda x.\, 2 \rceil : \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\stackrel{\text{def}}{=} & \bullet \rhd ((\lambda x{:}\, \star\, .\, 2 : \mathtt{num} \xRightarrow{+n} \star) : \star \to \star \xRightarrow{+o} \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\longrightarrow^{6} & X{:=}\mathtt{num} \rhd ((\lambda x{:}\, \star\, .\, 2 : \mathtt{num} \xRightarrow{+n} \star)\ (1 : \mathtt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star)) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd \lambda x{:}\, \star\, .\, 2 : \mathtt{num} \xRightarrow{+n} \star \xRightarrow{+\ell} X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd \mathtt{blame}\ {+\ell}
\end{array}$$

Example 7

$$\begin{array}{ll}
 & \bullet \rhd (\lceil \lambda x.\, x + 1 \rceil : \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\stackrel{\text{def}}{=} & \bullet \rhd ((\lambda x{:}\, \star\, .\, ((x : \star \xRightarrow{+m} \mathtt{num}) + 1) : \mathtt{num} \xRightarrow{+n} \star) : \star \to \star \xRightarrow{+o} \star \xRightarrow{+\ell} \forall X. X \to X)\ \mathtt{num}\ 1 \\
\longrightarrow^{6} & X{:=}\mathtt{num} \rhd ((\lambda x{:}\, \star\, .\, ((x : \star \xRightarrow{+m} \mathtt{num}) + 1) : \mathtt{num} \xRightarrow{+n} \star)\ (1 : \mathtt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star)) : \star \xRightarrow{+\ell} X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd ((1 : \mathtt{num} \xRightarrow{-X} X \xRightarrow{-\ell} \star \xRightarrow{+m} \mathtt{num}) + 1) : \mathtt{num} \xRightarrow{+n} \star \xRightarrow{+\ell} X \xRightarrow{+X} \mathtt{num} \\
\longrightarrow & X{:=}\mathtt{num} \rhd \mathtt{blame}\ {+m}
\end{array}$$

Figure 3. Example, generators and conversion

- If M_1 is not a value, apply the induction hypothesis for M_1 to obtain a decomposition $\mathcal{E}''$ of M_1. Then we pick $\mathcal{E}' = (\mathcal{E}'' : A \xRightarrow{\phi} B)$.

- $\boxed{\dfrac{\Gamma \vdash M_1 : A \qquad \Gamma \vdash A \prec B}{\Gamma \vdash (M_1 : A \xRightarrow{p} B) : B}}$

  - If M_1 is a value V, we proceed by cases on $A \prec B$.

    1. $\iota \prec \iota$:
       Pick $\mathcal{E}' = \square$ and $M' = (V : \iota \xRightarrow{p} \iota)$.
       $$V : \iota \xRightarrow{p} \iota \longmapsto V$$
    2. $A_1 \to A_2 \prec B_1 \to B_2$:
       $(V : A_1 \to A_2 \xRightarrow{p} B_1 \to B_2)$ is a value.
    3. $A \prec \forall X. B'$:
       - If $A = \forall X. A'$, then
         $$V : \forall X. A' \xRightarrow{p} B \longmapsto (V\ \star) : A'[X{:=}\star] \xRightarrow{p} B$$
         Pick $\mathcal{E}' = \square$, $M' = V : \forall X. A' \xRightarrow{p} B$.
       - Otherwise, $(V : A \xRightarrow{p} \forall X. B')$ is a value.
    4. $\forall X. A' \prec B$:
       $$V : \forall X. A' \xRightarrow{p} B \longmapsto (V\ \star) : A'[X{:=}\star] \xRightarrow{p} B$$
       Pick $\mathcal{E}' = \square$, $M' = V : \forall X. A' \xRightarrow{p} B$.
    5. $X \prec X$:
       Pick $\mathcal{E}' = \square$ and $M' = (V : X \xRightarrow{p} X)$.
       $$(V : X \xRightarrow{p} X) \longmapsto V$$
    6. $A \prec \star$:
       - If $A = G$, then $(V : G \xRightarrow{p} \star)$ is a value.
       - If $A = \star$, then
         $$V : \star \xRightarrow{p} \star \longmapsto V$$
         Pick $\mathcal{E}' = \square$ and $M' = V : \star \xRightarrow{p} \star$.
       - If A is not ground and not $\star$, then
         $$V : A \xRightarrow{p} \star \longmapsto V : A \xRightarrow{p} G \xRightarrow{p} \star$$
         Pick $\mathcal{E}' = \square$ and $M' = V : A \xRightarrow{p} \star$.
    7. $\star \prec B$:
       By canonical forms, we have $V = (V' : G \xRightarrow{q} \star)$.
       - If $B = H$ and $G = H$, then
         $$V' : G \xRightarrow{q} \star \xRightarrow{p} G \longmapsto V'$$
         Pick $\mathcal{E}' = \square$ and $M' = V : \star \xRightarrow{p} G$.
       - If $B = H$ and $G \neq H$, then
         $$V' : G \xRightarrow{q} \star \xRightarrow{p} H \longmapsto \mathtt{blame}\ p$$

Pick $\mathcal{E}' = \Box$ and $M' = V : \star \xRightarrow{p} H$.

– If $B = \star$, then

$$V : \star \xRightarrow{p} \star \longmapsto V$$

Pick $\mathcal{E}' = \Box$ and $M' = V : \star \xRightarrow{p} \star$.

– If B is not ground and not $\star$, then

$$V : \star \xRightarrow{p} B \longmapsto V : \star \xRightarrow{p} G \xRightarrow{p} B$$

where $G \prec B$. Pick $\mathcal{E}' = \Box$ and $M' = V : \star \xRightarrow{p} B$.

▪ If M_1 is not a value, the induction hypothesis for M_1 gives us a decomposition $\mathcal{E}''$, so we pick $\mathcal{E}' = (\mathcal{E}'' : A \xRightarrow{p} B)$.

- $\boxed{\dfrac{A : \texttt{tp}}{\texttt{blame}\, p : A}}$
  We satisfy the second option, picking $\mathcal{E}' = \Box$.

□

Lemma 5 (Context Inversion)**.** *If $\Sigma \rhd \mathcal{E}[M] : A$, then $\Sigma \rhd M : B$ and $\Sigma \rhd \mathcal{E} : B \Rightarrow A$ for some B.*

Lemma 6 (Plug)**.** *If $\Sigma \rhd M : B$ and $\Sigma \rhd \mathcal{E} : B \Rightarrow A$, then $\Sigma \rhd \mathcal{E}[M] : A$.*

Proposition 7 (Type safety)**.**

1. *If $\Sigma \rhd M : A$ and $\Sigma \rhd M \longrightarrow \Sigma' \rhd M'$ then $\Sigma' \rhd M' : A$.*
2. *If $\Sigma \rhd M : A$ then either*
   - $M = V'$
   - $M = \texttt{blame}\, p'$
   - $\Sigma \rhd M \longrightarrow \Sigma' \rhd M'$

   where all primed variables are existentially quantified.

Proof.

1. The proof of the first part (preservation) is by cases on $\Sigma \rhd M \longrightarrow \Sigma' \rhd M'$.
   - $\boxed{\Sigma \rhd \mathcal{E}[M_1] \longrightarrow \Sigma \rhd \mathcal{E}[M_1']}$
     We have $\Sigma \vdash M_1 : B$ and $\Sigma \vdash \mathcal{E} : B \Rightarrow A$ for some B by Lemma 5. Then $\Sigma \vdash M_1' : B$ by Lemma 2 (subject reduction). We conclude that $\Sigma \vdash \mathcal{E}[M_1'] : A$ by Lemma 6.
   - $\boxed{\Sigma \rhd \mathcal{E}[\texttt{blame}\, p] \longrightarrow \Sigma \rhd \texttt{blame}\, p}$
     We immediately have $\Sigma \vdash \texttt{blame}\, p : A$.
   - $\boxed{\Sigma \rhd \mathcal{E}[\nu X{:=}A'.N] \longrightarrow \Sigma,\ X{:=}A \rhd \mathcal{E}[N]}$
     We have $\Sigma \rhd \nu X{:=}A'.N : B$ and $\Sigma \rhd \mathcal{E} : B \Rightarrow A$ by Lemma 5. So we also have $\Sigma, X{:=}A' \rhd N : B$. Then we weaken the context to get $\Sigma, X{:=}A' \rhd \mathcal{E} : B \Rightarrow A$ and conclude that $\Sigma, X{:=}A' \rhd \mathcal{E}[N] : A$ by Lemma 6.
2. The proof of the second part (progress) is a direct result of Decomposition (Lemma 4).

□

3. Blame Safety of λB

Figure 4 presents three subtyping relations. Positive and negative subtyping, $A <:^+ B$ and $A <:^- B$, characterise when a cast $A \Longrightarrow^p B$ can never result in blaming p or $-p$, respectively. Naive subtyping, $A <:_n B$, characterises when type A is more *precise* than type B. All three relations are reflexive, and positive and naive subtyping are anti-symmetric and transitive. All three relations imply compatibility: if $A <:^+ B$ or $A <:^- B$ or $A <:_n B$ then $A \prec B$. The definition of the subtyping rules is driven by the corresponding rules for cast reductions.

Proposition 8 (Blame safety)**.**

1. *If M safe p and $\Sigma \rhd M \longrightarrow \Sigma' \rhd M'$ then M' safe p.*
2. *If M safe p then $\Sigma \rhd M \not\longrightarrow \Sigma' \rhd \texttt{blame}\, p$.*

Proof. We proceed by cases on $\Sigma \rhd M \longrightarrow \Sigma' \rhd M'$.

1. $\Sigma \rhd \mathcal{E}[M_1] \longrightarrow \Sigma \rhd \mathcal{E}[N_1]$ and $M_1 \longmapsto N_1$.
   We proceed by cases on $M_1 \longmapsto N_1$. We focus our attention on reductions that involve casts.
   (a) $V : \iota \xRightarrow{w} \iota \longmapsto V$:
   We immediately have V safe p.
   (b) $(V : A{\to}B \xRightarrow{q} C{\to}D)\ W \longmapsto V\ (W : C \xRightarrow{-q} A) : B \xRightarrow{q} D$:
   - If $p \neq q$ and $p \neq -q$, then we immediately have $(C \xRightarrow{-q} A)$ safe p and $(B \xRightarrow{q} D)$ safe p.
   - If $p = q$, then $A{\to}B <:^+ C{\to}D$. So $C <:^- A$ and $B <:^+ D$. Therefore $(C \xRightarrow{-q} A)$ safe p and $(B \xRightarrow{q} D)$ safe p.
   - If $p = -q$, then $A{\to}B <:^- C{\to}D$. So $C <:^+ A$ and $B <:^- D$. Therefore $(C \xRightarrow{-q} A)$ safe p and $(B \xRightarrow{q} D)$ safe p.

   (c) $(V : A \xRightarrow{q} \forall X.B)\ X \longmapsto V : A \xRightarrow{q} B$:
   Because $(V : A \xRightarrow{q} \forall X.B)$ is a value, we know that $A \neq \forall X.A$.
   - If $p \neq q$ and $p \neq -q$, then $(A \xRightarrow{q} B)$ safe p.
   - If $p = q$, then $A <:^+ \forall X.B$. So $A <:^+ B$ and we conclude that $(A \xRightarrow{q} B)$ safe p.
   - If $p = -q$, then $A <:^- \forall X.B$. So $A <:^- B$ and we conclude that $(A \xRightarrow{q} B)$ safe p.

   (d) $V : \forall X.A \xRightarrow{q} B \longmapsto (V\ \star) : A[X{:=}\star] \xRightarrow{q} B$:
   - If $p \neq q$ and $p \neq -q$, then we have $(A[X{:=}\star] \xRightarrow{q} B)$ safe p.
   - If $p = q$, then $\forall X.A <:^+ B$.
     First, we show that $A[X{:=}\star] <:^+ B$, considering two cases. Suppose $B = \forall X.B'$ for some X, B'. We have $\forall X.A <:^+ \forall X.B'$ by one of the following two typing derivations.

     $$\dfrac{\dfrac{A[X{:=}\star] <:^+ B'}{\forall X.A <:^+ B'}}{\forall X.A <:^+ \forall X.B'} \qquad \dfrac{\dfrac{A[X{:=}\star] <:^+ B'}{A[X{:=}\star] <:^+ \forall X.B'}}{\forall X.A <:^+ \forall X.B'}$$

     In either case we have $A[X{:=}\star] <:^+ B'$, from which we can deduce $A[X{:=}\star] <:^+ B$.
     Suppose $B \neq \forall X.B'$. Then we immediately have $A[X{:=}\star] <:^+ B$.
     We may now conclude that $(A[X{:=}\star] \xRightarrow{q} B)$ safe p.
   - If $p = -q$, then $\forall X.A <:^- B$. So $A[X{:=}\star] <:^- B$ and therefore $(A[X{:=}\star] \xRightarrow{q} B)$ safe p.

   (e) $V : X \xRightarrow{q} X \longmapsto V$:
   We immediately have V safe p.
   (f) $V : \star \xRightarrow{q} \star \longmapsto V$:
   We immediately have V safe p.
   (g) $V : A \xRightarrow{q} \star \longmapsto V : A \xRightarrow{q} G \xRightarrow{q} \star$:
   - If $p \neq q$ and $p \neq -q$, then we have $(A \xRightarrow{q} G)$ safe p and $(G \xRightarrow{q} \star)$ safe p.
   - If $p = q$, then $A <:^+ \star$. We immediate have $G <:^+ \star$. To show $A <:^+ G$, we proceed by cases on G. If $G = \iota$, then $A = G = \iota$ and $\iota <:^+ \iota$. If $G = X$, then $A = X$ and $X <:^+ X$. If $G = \star{\to}\star$, then $A = A_1{\to}A_2$

Positive subtype $\boxed{A <:^+ B}$

$$\frac{}{\iota <:^+ \iota} \qquad \frac{C <:^- A \quad B <:^+ D}{A \to B <:^+ C \to D} \qquad \frac{\begin{array}{c}[X : \texttt{tp}]\\ A <:^+ B\end{array} \quad X \notin A}{A <:^+ \forall X.B} \qquad \frac{A[X{:=}\star] <:^+ B}{\forall X.A <:^+ B} \qquad \frac{X : \texttt{tp}}{X <:^+ X} \qquad \frac{A : \texttt{tp}}{A <:^+ \star}$$

Negative subtype $\boxed{A <:^- B}$

$$\frac{}{\iota <:^- \iota} \qquad \frac{C <:^+ A \quad B <:^- D}{A \to B <:^- C \to D} \qquad \frac{\begin{array}{c}[X : \texttt{tp}]\\ A <:^- B\end{array} \quad X \notin A}{A <:^- \forall X.B} \qquad \frac{A[X{:=}\star] <:^- B}{\forall X.A <:^- B} \qquad \frac{X : \texttt{tp}}{X <:^- X} \qquad \frac{A <:^- G}{A <:^- \star} \qquad \frac{B : \texttt{tp}}{\star <:^- B}$$

Naive subtype $\boxed{A <:_n B}$

$$\frac{}{\iota <:^- \iota} \qquad \frac{A <:_n C \quad B <:_n D}{A \to B <:_n C \to D} \qquad \frac{\begin{array}{c}[X : \texttt{tp}]\\ A <:_n B\end{array} \quad X \notin A}{A <:_n \forall X.B} \qquad \frac{A[X{:=}\star] <:_n B}{\forall X.A <:_n B} \qquad \frac{X : \texttt{tp}}{X <:_n X} \qquad \frac{A : \texttt{tp}}{A <:_n \star}$$

Figure 4. Subtyping

and we need to show that $A_1 \to A_2 <:^+ \star \to \star$. Indeed, $\star <:^- A_1$ and $A_2 <:^+ \star$.

- If $p = -q$, then $A <:^- \star$. So $A <:^- G$ and therefore $(A \overset{q}{\Longrightarrow} G)$ safe p. Also, we have $G <:^- \star$ because $G <:^- G$, and therefore $(G \overset{q}{\Longrightarrow} \star)$ safe p

(h) $V : \star \overset{q}{\Longrightarrow} A \longmapsto V : \star \overset{q}{\Longrightarrow} G \overset{q}{\Longrightarrow} A$:

- If $p \neq q$ and $p \neq -q$, then we have $(\star \overset{q}{\Longrightarrow} G)$ safe p and $(G \overset{q}{\Longrightarrow} A)$ safe p.
- If $p = q$, then $\star <:^+ A$. From the reduction rule, we have $U(A)$, but that contradicts $\star <:^+ A$, so this case is vacuously true.
- Suppose $p = -q$. We immediately have $\star <:^- G$ but still need to show that $G <:^- A$. From the reduction rule, we have $U(A)$ and $A \prec G$, so $A = A_1 \to A_2$ and $G = \star \to \star$. We have $\star \to \star <:^- A_1 \to A_2$ because $A_1 <:^+ \star$ and $\star <:^- A_2$.

2. $\Sigma \rhd \mathcal{E}[\texttt{blame}\, p] \longrightarrow \texttt{blame}\, p$.
   We have a contradiction with the premise M safe p because it is not the case that $\mathcal{E}[\texttt{blame}\, p]$ safe p.
3. $\Sigma \rhd \mathcal{E}[\nu X{:=}A.N] \longrightarrow \Sigma, X{:=}A \rhd \mathcal{E}[N]$
   From $\mathcal{E}[\nu X{:=}A.N]$ safe p we have $\mathcal{E}[N]$ safe p.

□

4. Type Safety of λK

Lemma 9 (Subject Reduction)**.** *If $\Delta \vdash M : A$ and $M \longmapsto N$, then $\Delta \vdash N : A$.*

Proof. The proof is by cases on $M \longmapsto N$. We skip the two standard cases and focus on the two cases unique to λK.

- Case $\boxed{\lceil \lfloor V \rfloor_\kappa \rceil^p_\kappa \longmapsto V}$:
  We have $\Delta \vdash \lceil \lfloor V \rfloor_\kappa \rceil^p_\kappa : A$, so $\Delta \vdash \kappa : \texttt{key}\langle A \rangle$ and $\Delta \vdash \lfloor V \rfloor_\kappa : \texttt{bits}$. Therefore we conclude that $\Delta \vdash V : A$.
- Case $\boxed{\lceil \lfloor V \rfloor_\kappa \rceil^p_{\kappa'} \longmapsto \texttt{blame}\, p}$:
  We immediately have $\Delta \vdash \texttt{blame}\, p : A$.

□

Lemma 10 (Decomposition)**.** *If $\Sigma \vdash M : A$, then either*

1. *$M = V'$,*
2. *$M = \mathcal{E}'[\texttt{blame}\, p']$,*
3. *$M = \mathcal{E}'[\texttt{new}\langle B \rangle]$, or*
4. *$M = \mathcal{E}'[M']$ and $M' \longmapsto N'$.*

where all primed variables are existentially quantified.

Proof. The proof is by induction on $M : A$.

- $\boxed{\texttt{new}\langle B \rangle : \texttt{key}\langle B \rangle}$
  Pick $\mathcal{E}' = \square$ to satisfy the third option.
- $\boxed{\lfloor M_1 \rfloor_{M_2}}$
  - Suppose M_1 and M_2 are values. Let $V_1 = M_1$. By canonical forms, $M_2 = \kappa$. Then pick $V' = \lfloor V_1 \rfloor_\kappa$.
  - Suppose M_1 is a value V_1 but M_2 is not a value. Then we apply the induction hypothesis for M_2 to obtain $\mathcal{E}''$ then pick $\mathcal{E}' = \lfloor V_1 \rfloor_{\mathcal{E}'}$.
  - Suppose M_1 is not a value. Then we apply the induction hypothesis for M_1 to obtain $\mathcal{E}''$ then pick $\mathcal{E}' = \lfloor \mathcal{E}' \rfloor_{M_2}$.
- $\boxed{\lceil M_1 \rceil^p_{M_2}}$
  - Suppose M_1 and M_2 are values. By canonical forms, $M_1 = \lfloor V' \rfloor_\kappa$ and $M_2 = \kappa'$. We satisfy the fourth option by picking $\mathcal{E}' = \square$ and $M' = \lceil \lfloor V' \rfloor_\kappa \rceil^p_{\kappa'}$. If $\kappa = \kappa'$ we have
  $$\lceil \lfloor V' \rfloor_\kappa \rceil^p_{\kappa'} \longmapsto V'$$
  Otherwise we have
  $$\lceil \lfloor V' \rfloor_\kappa \rceil^p_{\kappa'} \longmapsto \texttt{blame}\, p$$
  - Suppose M_1 is a value V_1 but M_2 is not a value. Then we apply the induction hypothesis for M_2 to obtain $\mathcal{E}''$ then pick $\mathcal{E}' = \lceil V_1 \rceil^p_{\mathcal{E}''}$.
  - Suppose M_1 is not a value. Then we apply the induction hypothesis for M_1 to obtain $\mathcal{E}''$ then pick $\mathcal{E}' = \lceil \mathcal{E}'' \rceil^p_{M_2}$.

The rest of the cases are standard. □

Proposition 11 (Type safety)**.**

1. *If $\Delta \rhd M : A$ and $\Delta \rhd M \longrightarrow \Delta' \rhd M'$ then $\Delta' \rhd M' : A$.*
2. *If $\Sigma \rhd M : A$ then either*
   - *$M = V'$*
   - *$M = \texttt{blame}\, p'$*
   - *$\Sigma \rhd M \longrightarrow \Sigma' \rhd M'$*

   where all primed variables are existentially quantified.

Proof.

1. The proof of the first part (preservation) is by cases on $\Delta \rhd M \longrightarrow \Delta' \rhd M'$.

- $\boxed{\dfrac{M_1 \longrightarrow N}{\Delta \rhd \mathcal{E}[M_1] \longrightarrow \Delta \rhd \mathcal{E}[N]}}$
  We have $\Delta \rhd M_1 : B$ and $\Delta \rhd \mathcal{E} : B \Longrightarrow A$ for some B. So $\Delta \rhd N : B$ by subject reduction (Lemma 9). Finally, we have $\Delta \rhd \mathcal{E}[N]$.
- $\boxed{\dfrac{}{\Delta \rhd \mathcal{E}[\texttt{blame}\, p] \longrightarrow \Delta \rhd \texttt{blame}\, p}}$
  We immeidately have $\Delta \vdash \texttt{blame}\, p : A$.
- $\boxed{\dfrac{\kappa \notin \Delta}{\Delta \rhd \mathcal{E}[\texttt{new}\langle B\rangle] \longrightarrow \Delta,\, \kappa : \texttt{key}\langle B\rangle \rhd \mathcal{E}[\kappa]}}$
  We have $\Delta \vdash \texttt{new}\langle B\rangle : \texttt{key}\langle B\rangle$, so $\Delta \vdash \mathcal{E} : \texttt{key}\langle B\rangle \Rightarrow A$. Thus, $\Delta, \kappa : \texttt{key}\langle B\rangle \vdash \mathcal{E}[\kappa] : A$.

2. The proof of the second part (progress) is a direct result of Decomposition (Lemma 10).

□

5. Results for Translation of λB to λK

$$\Gamma ::= \bullet \mid \Gamma,\, x{:=}A \mid \Gamma,\, X \mid \Gamma,\, X{:=}A$$

$\boxed{\Gamma \sim R}$

$$\bullet \sim R \qquad \frac{\Gamma \sim R \quad X \mapsto (\bullet, x_k) \in R}{(\Gamma,\, X) \sim R}$$

$$\frac{\Gamma \sim R \quad X \mapsto (x_j, x_k) \in R)}{(\Gamma,\, X{:=}A) \sim R}$$

$$\frac{\Gamma \sim R}{(\Gamma,\, x{:=}A) \sim R}$$

$$\Sigma \sim R \equiv \forall X{:=}A \in \Sigma. \exists \kappa_j \kappa_k.\ X \mapsto (\kappa_j, \kappa_k) \in R$$

$\boxed{\llparenthesis \Gamma \rrparenthesis_R}$

$$\begin{aligned}
\llparenthesis \bullet \rrparenthesis_R &= \bullet \\
\llparenthesis \Gamma, x{:=}A \rrparenthesis_R &= \llparenthesis \Gamma \rrparenthesis_R, x{:=}\llparenthesis A \rrparenthesis \\
\llparenthesis \Gamma, X \rrparenthesis_R &= \llparenthesis \Gamma \rrparenthesis_R, x_k : \texttt{key}\langle\texttt{bits}\rangle \\
&\quad \text{if } X \mapsto (\bullet, x_k) \in R \\
\llparenthesis \Gamma, X{:=}A \rrparenthesis_R &= \llparenthesis \Gamma \rrparenthesis_R, x_j : \texttt{key}\langle\llparenthesis A \rrparenthesis\rangle, x_k ; \texttt{key}\langle\texttt{bits}\rangle \\
&\quad \text{if } X \mapsto (x_j, x_k) \in R
\end{aligned}$$

$\boxed{\llparenthesis \Sigma \rrparenthesis_R}$

$$\begin{aligned}
\llparenthesis \bullet \rrparenthesis_R &= \bullet \\
\llparenthesis \Sigma, X{:=}A \rrparenthesis_R &= \llparenthesis \Sigma \rrparenthesis, \kappa_j : \texttt{key}\langle\llparenthesis A \rrparenthesis\rangle, \kappa_k : \texttt{key}\langle\texttt{bits}\rangle \\
&\quad \text{if } X \mapsto (\kappa_j, \kappa_k) \in R
\end{aligned}$$

Lemma 12. *If* $\Sigma; \Gamma \vdash A \prec^{\phi} B$, $\Sigma \sim R$, *and* $\Gamma \sim R$, *then* $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis A \xrightarrow{\phi} B \rrparenthesis_R : \llparenthesis A \rrparenthesis \rightarrow \llparenthesis B \rrparenthesis$.

Proof. by induction on the derivation of $\Sigma; \Gamma \vdash A \prec^{\phi} B$.

- Case $\boxed{\iota \prec^{\phi} \iota}$ straightforward
- Case $\boxed{A_1{\rightarrow}A_2 \prec^{\phi} B_1{\rightarrow}B_2}$
  follows immediately from the two induction hypotheses.
- Case $\boxed{\forall X.A \prec^{\phi} \forall X.B}$
  We have
  $$\llparenthesis \forall X.A \xRightarrow{\phi} \forall X.B \rrparenthesis_R = \begin{array}{l} \lambda v{:}\texttt{key}\langle\texttt{bits}\rangle{\rightarrow}\llparenthesis A \rrparenthesis. \\ \lambda k{:}\texttt{key}\langle\texttt{bits}\rangle.\, (v\ k)\texttt{@} \\ \llparenthesis A \xRightarrow{\phi} B \rrparenthesis_{R'} \end{array}$$
  where $R' = (R, X \mapsto (\bullet, k))$ and we need to show that the RHS has type
  $$\llparenthesis \forall X.A \rrparenthesis \rightarrow \llparenthesis \forall X.B \rrparenthesis$$
  So it suffices to show that
  $$\llparenthesis \Sigma \rrparenthesis_{R'}; \llparenthesis \Gamma, X \rrparenthesis_{R'} \vdash \llparenthesis A \xRightarrow{\phi} B \rrparenthesis_{R'} : \llparenthesis B \rrparenthesis$$
  which we obtain by the induction hypothesis, noting that $(\Gamma, X) \sim (R, X \mapsto (\bullet, k))$.
- Case $\boxed{X \prec^{+X} B}$
  Because $X := B \in \Gamma$ or $X := B \in \Sigma$, we have $R(X) = (k, j)$ with $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash j : \texttt{key}\langle\llparenthesis B \rrparenthesis\rangle$. We also have
  $$\llparenthesis X \xRightarrow{+X} B \rrparenthesis_R = \lambda v{:}\texttt{bits}.\, \lceil v \rceil_j^{\bullet}$$
  We conclude that
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash (\lambda v{:}\texttt{bits}.\, \lceil v \rceil_j^{\bullet}) : \texttt{bits} \rightarrow \llparenthesis B \rrparenthesis$$
- Case $\boxed{A \prec^{-X} X}$ Because $X := A \in \Gamma$ or $X := A \in \Sigma$, we have $R(X) = (k, j)$ with $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash j : \texttt{key}\langle\llparenthesis A \rrparenthesis\rangle$. So
  $$\llparenthesis A \xRightarrow{-X} X \rrparenthesis_R = \lambda v{:}\llparenthesis A \rrparenthesis.\, \lfloor v \rfloor_j$$
  We conclude that
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash (\lambda v{:}\llparenthesis A \rrparenthesis.\, \lfloor v \rfloor_j) : \llparenthesis A \rrparenthesis \rightarrow \texttt{bits}$$
- Case $\boxed{X \prec^{\phi} X}$ straightforward
- Case $\boxed{\star \prec^{\phi} \star}$ straightforward

□

Lemma 13. *If* $\Sigma; \Gamma \vdash G$, $\Sigma \sim R$, *and* $\Gamma \sim R$, *then* $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis G \rrparenthesis_R : \texttt{key}\langle\llparenthesis G \rrparenthesis\rangle$.

Lemma 14. *If* $\Sigma; \Gamma \vdash A \prec B$, $\Sigma \sim R$, *and* $\Gamma \sim R$, *then* $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis A \xRightarrow{p} B \rrparenthesis_R : \llparenthesis A \rrparenthesis \rightarrow \llparenthesis B \rrparenthesis$.

Proof. The proof is by induction on the sum of the size of A and B. Then we proceed by cases on $\Sigma; \Gamma \vdash A \prec B$.

- Case $\boxed{\Sigma; \Gamma \vdash A \prec \star}$
  Suppose $A = G$. We have
  $$\llparenthesis G \xRightarrow{p} \star \rrparenthesis_R = \lambda v : \llparenthesis G \rrparenthesis.\, \lfloor v \rfloor_k$$
  Where $k = \llparenthesis G \rrparenthesis_R$. We have
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash k : \texttt{key}\langle\llparenthesis G \rrparenthesis\rangle$$
  by Lemma 13. Therefore
  $$\vdash (\lambda v : \llparenthesis G \rrparenthesis.\, \lfloor v \rfloor_k) : \llparenthesis G \rrparenthesis \rightarrow \texttt{bits}$$
  Suppose $A \neq G$. We have
  $$\llparenthesis A \xRightarrow{p} \star \rrparenthesis_R = \lambda v : \llparenthesis A \rrparenthesis.\, (v \mathbin{\texttt{@}} \llparenthesis A \xRightarrow{p} G \rrparenthesis_R) \mathbin{\texttt{@}} \llparenthesis G \xRightarrow{p} \star \rrparenthesis_R$$
  and $A = A_1 \rightarrow A_2$ and $G = \star \rightarrow \star$. We use the induction hypothesis for $\llparenthesis \star \xRightarrow{-p} A_1 \rrparenthesis_R$ and $\llparenthesis A_2 \xRightarrow{p} \star \rrparenthesis_R$ to obtain
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis A \xRightarrow{p} G \rrparenthesis_R : \llparenthesis A \rrparenthesis \rightarrow \llparenthesis G \rrparenthesis$$
  Also, by the same reasoning as in the $A = G$ case, we have
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis G \xRightarrow{p} \star \rrparenthesis_R : \llparenthesis G \rrparenthesis \rightarrow \llparenthesis \star \rrparenthesis$$
  Therefore
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \begin{array}{l} \lambda v : \llparenthesis A \rrparenthesis.\, (v \mathbin{\texttt{@}} \llparenthesis A \xRightarrow{p} G \rrparenthesis_R) \mathbin{\texttt{@}} \llparenthesis G \xRightarrow{p} \star \rrparenthesis_R \\ : \llparenthesis A \rrparenthesis \rightarrow \llparenthesis \star \rrparenthesis \end{array}$$

- Case $\boxed{\Sigma; \Gamma \vdash \star \prec A}$
  Suppose $A = G$. We have
  $$\llparenthesis \star \xRightarrow{p} G \rrparenthesis_R = \lambda v{:}\mathtt{bits}.\ \lceil v \rceil^p_k$$
  where $k = \llparenthesis G \rrparenthesis_R$. We have
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R, v{:}\mathtt{bits} \vdash k : \mathtt{key}\langle \llparenthesis G \rrparenthesis \rangle$$
  by Lemma 13. Therefore
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \lambda v{:}\mathtt{bits}.\ \lceil v \rceil^p_k : \mathtt{bits} \to \llparenthesis G \rrparenthesis$$
  Suppose $A \neq G$. We have
  $$\llparenthesis \star \xRightarrow{p} A \rrparenthesis_R = \lambda v{:}\mathtt{bits}.\ (v \ \mathtt{@}\ \llparenthesis \star \xRightarrow{p} G \rrparenthesis_R)\ \mathtt{@}\ \llparenthesis G \xRightarrow{p} A \rrparenthesis_R$$
  We have
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R, v{:}\mathtt{bits} \vdash \llparenthesis \star \xRightarrow{p} G \rrparenthesis_R : \mathtt{bits} \to \llparenthesis G \rrparenthesis$$
  by the same reasoning as in the case for $A = G$ followed by weakening. We have $G = \star{\to}\star$ and $A = A_1{\to}A_2$. We use the induction hypothesis for $\llparenthesis A_1 \xRightarrow{-p} \star \rrparenthesis_R$ and $\llparenthesis \star \xRightarrow{p} A_2 \rrparenthesis_R$ and weakening to obtain
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R, v{:}\mathtt{bits} \vdash \llparenthesis G \xRightarrow{p} A \rrparenthesis_R : \llparenthesis G \rrparenthesis \to \llparenthesis A \rrparenthesis$$
  We conclude that
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \lambda v{:}\mathtt{bits}.\ (v \mathtt{@} \llparenthesis \star \xRightarrow{p} G \rrparenthesis_R) \mathtt{@} \llparenthesis G \xRightarrow{p} A \rrparenthesis_R : \llparenthesis A \rrparenthesis$$
- $\boxed{\Sigma; \Gamma \vdash A \prec \forall X. B'[X]}$
  We have
  $$\llparenthesis A \xRightarrow{p} \forall X. B'[X] \rrparenthesis_R = \begin{array}{l} \lambda v{:}\llparenthesis A \rrparenthesis.\ \lambda k{:}\mathtt{key}\langle\mathtt{bits}\rangle. \\ v \ \mathtt{@}\ \llparenthesis A \xRightarrow{p} B'[X] \rrparenthesis_{R'} \end{array}$$
  where $R' = (R, X \mapsto (\bullet, k))$. We need to show that the RHS has type
  $$\llparenthesis A \rrparenthesis \to \mathtt{key}\langle\mathtt{bits}\rangle \to \llparenthesis B'[X] \rrparenthesis$$
  Let $\Gamma' = (\Gamma, X)$ and note that $\Gamma' \sim R'$ and $\Sigma \sim R'$. So by the induction hypothesis, we have
  $$\llparenthesis \Sigma \rrparenthesis_{R'}; \llparenthesis \Gamma' \rrparenthesis_{R'} \vdash \llparenthesis A \xRightarrow{p} B'[X] \rrparenthesis_{R'} : \llparenthesis A \rrparenthesis \to \llparenthesis B'[X] \rrparenthesis$$
  Thus it is straightforward to conclude this case.
- $\boxed{\Sigma; \Gamma \vdash \forall X. A' \prec B}$ We have
  $$\llparenthesis \forall X. A' \xRightarrow{p} B \rrparenthesis_R = \begin{array}{l} \lambda v{:}\mathtt{key}\langle\mathtt{bits}\rangle{\to}\llparenthesis A' \rrparenthesis. \\ \mathtt{let}\ j = \mathtt{new}\langle\mathtt{bits}\rangle\ \mathtt{in} \\ \mathtt{let}\ k = \mathtt{new}\langle\mathtt{bits}\rangle\ \mathtt{in} \\ ((v\ k)\ \mathtt{@}\ \llparenthesis A' \xRightarrow{+X} A'[X{:=}\star] \rrparenthesis_{R'}) \\ \mathtt{@}\llparenthesis A'[X{:=}\star] \xRightarrow{p} B \rrparenthesis_R \end{array}$$
  where $R' = (R, X \mapsto (j, k))$. We need to show that the RHS has type
  $$(\mathtt{key}\langle\mathtt{bits}\rangle \to \llparenthesis A' \rrparenthesis) \to \llparenthesis B \rrparenthesis$$
  By Lemma 12 we have the following, noting that $\Sigma \sim R'$ and $\Gamma' \sim R'$, where $\Gamma' = (\Gamma, X{:=}\star)$.
  $$\llparenthesis \Sigma \rrparenthesis; \llparenthesis \Gamma' \rrparenthesis \vdash \llparenthesis A' \xRightarrow{+X} A'[X{:=}\star] \rrparenthesis_{R'} : \llparenthesis A' \rrparenthesis \to \llparenthesis A'[X{:=}\star] \rrparenthesis$$
  By the induction hypthesis, we have
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis A'[X{:=}\star] \xRightarrow{p} B \rrparenthesis_R : \llparenthesis A'[X{:=}\star] \rrparenthesis \to \llparenthesis B \rrparenthesis$$
  We then conclude this case with some uses of weakening and the typing rule for function application.
- $\boxed{\Sigma; \Gamma \vdash A_1{\to}A_2 \prec B_1{\to}B_2}$
  Use the induction hypotheses for $\llparenthesis B_1 \xRightarrow{-p} A_1 \rrparenthesis_R$ and $\llparenthesis A_2 \xRightarrow{p} B_2 \rrparenthesis_R$.
- $\boxed{\Sigma; \Gamma \vdash \iota \prec \iota}$ straightforward
- $\boxed{\Sigma; \Gamma \vdash X \prec X}$ straightforward
- $\boxed{\Sigma; \Gamma \vdash \star \prec \star}$ straightforward

□

Proposition 15 (Type preservation, λB to λK)**.** *If* $\Sigma; \Gamma \rhd M : A$, $\Sigma \sim R$, *and* $\Gamma \sim R$, *then* $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \rhd \llparenthesis M \rrparenthesis_R : \llparenthesis A \rrparenthesis$.

Proof. The proof is by induction on $\Sigma; \Gamma \rhd M : A$.

- $\boxed{\llparenthesis \Lambda X.\ V[X] \rrparenthesis_R = \lambda k : \mathtt{key}\langle\mathtt{bits}\rangle.\ \llparenthesis V[X] \rrparenthesis_{R,\ X \mapsto (\bullet, k)}}$
  We have $A = \forall X. A'$ and $\Sigma; \Gamma, X \vdash V[X] : A'$. We need to show
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash (\lambda k : \mathtt{key}\langle\mathtt{bits}\rangle.\ \llparenthesis V[X] \rrparenthesis_{R,\ X \mapsto (\bullet, k)}) : \llparenthesis A \rrparenthesis$$
  where $\llparenthesis A \rrparenthesis = \mathtt{key}\langle\mathtt{bits}\rangle \to \llparenthesis A' \rrparenthesis$. Let $R' = R,\ X \mapsto (\bullet, k)$, We apply the induction hypothesis for $V[X]$, noting that $\Sigma \sim R'$ and $(\Gamma, X) \sim R'$.
  $$\llparenthesis \Sigma \rrparenthesis_{R'}; \llparenthesis \Gamma, X \rrparenthesis_{R'} \vdash \llparenthesis V[X] \rrparenthesis_{R'} : \llparenthesis A' \rrparenthesis$$
  so
  $$\llparenthesis \Sigma \rrparenthesis_{R'}; \llparenthesis \Gamma \rrparenthesis, k : \mathtt{key}\langle\mathtt{bits}\rangle \vdash \llparenthesis V[X] \rrparenthesis_{R'} : \llparenthesis A' \rrparenthesis$$
  thus
  $$\llparenthesis \Sigma \rrparenthesis_{R'}; \llparenthesis \Gamma \rrparenthesis_{R'} \vdash \lambda k : \mathtt{key}\langle\mathtt{bits}\rangle.\ \llparenthesis V[X] \rrparenthesis_{R'} : \llparenthesis A \rrparenthesis$$
  and we conclude because $\llparenthesis \Sigma \rrparenthesis_{R'} = \llparenthesis \Sigma \rrparenthesis_R$ and $\llparenthesis \Gamma \rrparenthesis_{R'} = \llparenthesis \Gamma \rrparenthesis_R$.
- $\boxed{\llparenthesis L\ X \rrparenthesis_R = \llparenthesis L \rrparenthesis_R\ \underline{k} \text{ and } (X \mapsto (j, k)) \in R}$
  We have $\Sigma; \Gamma \vdash L : \forall X. B$ and $A = B$. Also, either $X{:=}A \in \Gamma$ or $X{:=}A \in \Sigma$.
  By the induction hypothesis for L we have
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis L \rrparenthesis_R : \mathtt{key}\langle\mathtt{bits}\rangle \to \llparenthesis B \rrparenthesis$$
  Next, we show that $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash k : \mathtt{key}\langle\mathtt{bits}\rangle$, considering two cases.
  - If $X{:=}A \in \Gamma$, then $k : \mathtt{key}\langle\mathtt{bits}\rangle \in \llparenthesis \Gamma \rrparenthesis_R$ (because $\Gamma \sim R$) and our goal follows.
  - If $X{:=}A \in \Sigma$, then $k : \mathtt{key}\langle\mathtt{bits}\rangle \in \llparenthesis \Sigma \rrparenthesis_R$ (because $\Sigma \sim R$) and our goal follows.

  Therefore
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis L \rrparenthesis_R\ k : \llparenthesis B \rrparenthesis$$
- $\boxed{\llparenthesis \nu X{:=}B. N \rrparenthesis_R = \left(\begin{array}{l} \mathtt{let}\ j = \mathtt{new}\langle \llparenthesis B \rrparenthesis \rangle\ \mathtt{in} \\ \mathtt{let}\ k = \mathtt{new}\langle\mathtt{bits}\rangle\ \mathtt{in} \\ \llparenthesis N \rrparenthesis_{R,\ X \mapsto (j,k)} \end{array} \right)}$
  We have $\Sigma; \Gamma, X{:=}B \vdash N : A$. Let $R' = R[X \mapsto (j, k)]$. By the induction hypothesis we have the following, noting that $\Sigma \sim R'$ and $(\Gamma, X{:=}B) \sim R'$.
  $$\llparenthesis \Sigma \rrparenthesis_{R'}; \llparenthesis \Gamma, X{:=}B \rrparenthesis_{R'} \vdash \llparenthesis N \rrparenthesis_{R'} : \llparenthesis A \rrparenthesis$$
  so
  $$\llparenthesis \Sigma \rrparenthesis_{R'}; \llparenthesis \Gamma \rrparenthesis_{R'}, j : \mathtt{key}\langle \llparenthesis B \rrparenthesis \rangle, k : \mathtt{key}\langle\mathtt{bits}\rangle \vdash \llparenthesis N \rrparenthesis_{R'} : \llparenthesis A \rrparenthesis$$
  and we conclude that
  $$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \left(\begin{array}{l} \mathtt{let}\ j = \mathtt{new}\langle \llparenthesis B \rrparenthesis \rangle\ \mathtt{in} \\ \mathtt{let}\ k = \mathtt{new}\langle\mathtt{bits}\rangle\ \mathtt{in} \\ \llparenthesis N \rrparenthesis_{R,\ X \mapsto (j,k)} \end{array} \right) : \llparenthesis A \rrparenthesis$$
  noting that $\llparenthesis \Sigma \rrparenthesis_{R'} = \llparenthesis \Sigma \rrparenthesis_R$ and $\llparenthesis \Gamma \rrparenthesis_{R'} = \llparenthesis \Gamma \rrparenthesis_R$.

- $\boxed{\llparenthesis M_1 : A_1 \stackrel{\phi}{\Longrightarrow} A \rrparenthesis_R = \llparenthesis M_1 \rrparenthesis_R \mathbin{@} \llparenthesis A_1 \stackrel{\phi}{\Longrightarrow} A \rrparenthesis_R}$
  We have $\Sigma;\Gamma \vdash M_1 : A_1$ and $A_1 \prec^{\phi} A$. By the induction hypothesis, $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis M_1 \rrparenthesis_R : \llparenthesis A_1 \rrparenthesis$. By Lemma 12, $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis A_1 \stackrel{\phi}{\Longrightarrow} A \rrparenthesis_R : \llparenthesis A_1 \rrparenthesis \to \llparenthesis A \rrparenthesis$. Therefore

$$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis M_1 \rrparenthesis_R \mathbin{@} \llparenthesis A_1 \stackrel{\phi}{\Longrightarrow} A \rrparenthesis_R : \llparenthesis A \rrparenthesis$$

- $\boxed{\llparenthesis M_1 : A_1 \stackrel{p}{\Longrightarrow} A \rrparenthesis_R = \llparenthesis M_1 \rrparenthesis_R \mathbin{@} \llparenthesis A_1 \stackrel{p}{\Longrightarrow} A \rrparenthesis_R}$
  We have $\Sigma;\Gamma \vdash M_1 : A_1$ and $A_1 \prec A$. By the induction hypothesis, $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis M_1 \rrparenthesis_R : \llparenthesis A_1 \rrparenthesis$. By Lemma 14, $\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis A_1 \stackrel{p}{\Longrightarrow} A \rrparenthesis_R : \llparenthesis A_1 \rrparenthesis \to \llparenthesis A \rrparenthesis$. Therefore

$$\llparenthesis \Sigma \rrparenthesis_R; \llparenthesis \Gamma \rrparenthesis_R \vdash \llparenthesis M_1 \rrparenthesis_R \mathbin{@} \llparenthesis A_1 \stackrel{p}{\Longrightarrow} A \rrparenthesis_R : \llparenthesis A \rrparenthesis$$

□

Lemma 16. *If $M \longmapsto M'$ then $\llparenthesis M \rrparenthesis_R \longmapsto^{?} \llparenthesis M' \rrparenthesis_R$.*

Proof. We proceed by cases on $M \longmapsto M'$.

- $\boxed{(\Lambda X.\, V[X])\ X \longmapsto_{\mathsf{B}} V[X]}$ Note that $R(X) = (j,k)$. We have

$$\begin{aligned} \llparenthesis M \rrparenthesis_R &= \llparenthesis (\Lambda X.\, V[X])\ X \rrparenthesis_R \\ &= (\lambda x_k.\, \llparenthesis V[X] \rrparenthesis_{R, X \mapsto (\bullet, x_k)})\ k \\ &\longmapsto_{\mathsf{K}} \llparenthesis V[X] \rrparenthesis_{R, X \mapsto (\bullet, k)} = \llparenthesis M' \rrparenthesis_R \end{aligned}$$

  Because $\llparenthesis V[X] \rrparenthesis_{R, X \mapsto (\bullet, k)} = \llparenthesis V[X] \rrparenthesis_R$, this case is finished.

- $\boxed{V : \iota \stackrel{\phi}{\Longrightarrow} \iota \longmapsto_{\mathsf{B}} V}$ We have

$$\llparenthesis M \rrparenthesis_R = \llparenthesis V \rrparenthesis_R \mathbin{@} \llparenthesis \iota \stackrel{\phi}{\Longrightarrow} \iota \rrparenthesis_R = \llparenthesis V \rrparenthesis_R \mathbin{@} (\overline{\lambda} v{:}\iota.\, v) = \llparenthesis M' \rrparenthesis_R$$

- $\boxed{(V{:}A{\to}B \stackrel{\phi}{\Longrightarrow} C{\to}D)\ V' \longmapsto_{\mathsf{B}} V\ (V'{:}C \stackrel{-\phi}{\Longrightarrow} A){:}B \stackrel{\phi}{\Longrightarrow} D}$
  We have

$$\begin{aligned} &\llparenthesis A{\to}B \stackrel{\phi}{\Longrightarrow} C{\to}D \rrparenthesis_R = \\ &\quad \overline{\lambda} v.\, \lambda w.\, (v\ (w \mathbin{@} \llparenthesis C \stackrel{-\phi}{\Longrightarrow} A \rrparenthesis_R)) \mathbin{@} \llparenthesis B \stackrel{\phi}{\Longrightarrow} D \rrparenthesis_R \end{aligned}$$

  So

$$\begin{aligned} \llparenthesis M \rrparenthesis_R &= \llparenthesis (V{:}A{\to}B \stackrel{\phi}{\Longrightarrow} C{\to}D)\ V' \rrparenthesis_R \\ &= (\llparenthesis V \rrparenthesis_R{:}\llparenthesis A{\to}B \stackrel{\phi}{\Longrightarrow} C{\to}D \rrparenthesis_R)\ \llparenthesis V' \rrparenthesis_R \\ &\longmapsto_{\mathsf{K}} (\llparenthesis V \rrparenthesis_R\ (\llparenthesis V' \rrparenthesis_R \mathbin{@} \llparenthesis C \stackrel{-\phi}{\Longrightarrow} A \rrparenthesis_R)) \mathbin{@} \llparenthesis B \stackrel{\phi}{\Longrightarrow} D \rrparenthesis_R \end{aligned}$$

- $\boxed{(V : \forall X.A \stackrel{\phi}{\Longrightarrow} \forall X.B)\ X \longmapsto V\ X : A \stackrel{\phi}{\Longrightarrow} B}$
  We have

$$\begin{aligned} &\llparenthesis \forall X.A \stackrel{\phi}{\Longrightarrow} \forall X.B \rrparenthesis_R = \\ &\quad \overline{\lambda} v.\, \lambda k'.\, (v\ k') \mathbin{@} \llparenthesis A \stackrel{\phi}{\Longrightarrow} B \rrparenthesis_{R, X \mapsto (\bullet, k')} \end{aligned}$$

  and $R(X) = (j,k)$. So

$$\begin{aligned} \llparenthesis M \rrparenthesis_R &= (\llparenthesis V \rrparenthesis_R\ k) \mathbin{@} \llparenthesis A \stackrel{\phi}{\Longrightarrow} B \rrparenthesis_{R, X \mapsto (\bullet, k)} \\ &= (\llparenthesis V \rrparenthesis_R\ k) \mathbin{@} \llparenthesis A \stackrel{\phi}{\Longrightarrow} B \rrparenthesis_R \\ &= \llparenthesis V\ X : A \stackrel{\phi}{\Longrightarrow} B \rrparenthesis_R \\ &= \llparenthesis M' \rrparenthesis_R \end{aligned}$$

- $\boxed{V : X \stackrel{\phi}{\Longrightarrow} X \longmapsto V}$
  We have

$$\llparenthesis X \stackrel{\phi}{\Longrightarrow} X \rrparenthesis_R = \overline{\lambda} v.\, v$$

  So

$$\llparenthesis M \rrparenthesis_R = ((\llparenthesis V \rrparenthesis_R \mathbin{@} \overline{\lambda} v.\, v) = \llparenthesis V \rrparenthesis_R = \llparenthesis M' \rrparenthesis_R$$

- $\boxed{V : A \stackrel{-X}{\Longrightarrow} X \stackrel{+X}{\Longrightarrow} A \longmapsto V}$
  We have

$$\begin{aligned} \llparenthesis A \stackrel{-X}{\Longrightarrow} X \rrparenthesis_R &= \overline{\lambda} v.\, \lfloor v \rfloor_j \\ \llparenthesis X \stackrel{+X}{\Longrightarrow} A \rrparenthesis_R &= \overline{\lambda} v.\, \lceil v \rceil_j^{\bullet} \end{aligned}$$

  By Proposition 17 (part 1), $\llparenthesis V \rrparenthesis_R$ is a value. So

$$\llparenthesis M \rrparenthesis_R = \lceil \lfloor \llparenthesis V \rrparenthesis_R \rfloor_j \rceil_j^{\bullet} \longmapsto_{\mathsf{K}} \llparenthesis V \rrparenthesis_R = \llparenthesis M' \rrparenthesis_R$$

- $\boxed{V : \star \stackrel{\phi}{\Longrightarrow} \star \longmapsto_{\mathsf{B}} V}$ straightforward
- $\boxed{V : \iota \stackrel{p}{\Longrightarrow} \iota \longmapsto_{\mathsf{B}} V}$ straightforward
- $\boxed{\begin{array}{l} (V : A \to B \stackrel{p}{\Longrightarrow} C \to D)\ W \\ \longmapsto_{\mathsf{B}} V\ (W : C \stackrel{-p}{\Longrightarrow} A) : B \stackrel{p}{\Longrightarrow} D \end{array}}$
  We have

$$\begin{aligned} &\llparenthesis A{\to}B \stackrel{p}{\Longrightarrow} C{\to}D \rrparenthesis_R = \\ &\quad \overline{\lambda} v.\, \lambda w.\, (v\ (w \mathbin{@} \llparenthesis C \stackrel{-\phi}{\Longrightarrow} A \rrparenthesis_R)) \mathbin{@} \llparenthesis B \stackrel{p}{\Longrightarrow} D \rrparenthesis_R \end{aligned}$$

  So

$$\begin{aligned} \llparenthesis M \rrparenthesis_R &= \llparenthesis (V{:}A{\to}B \stackrel{p}{\Longrightarrow} C{\to}D)\ V' \rrparenthesis_R \\ &= (\llparenthesis V \rrparenthesis_R{:}\llparenthesis A{\to}B \stackrel{p}{\Longrightarrow} C{\to}D \rrparenthesis_R)\ \llparenthesis V' \rrparenthesis_R \\ &\longmapsto_{\mathsf{K}} (\llparenthesis V \rrparenthesis_R\ (\llparenthesis V' \rrparenthesis_R \mathbin{@} \llparenthesis C \stackrel{-\phi}{\Longrightarrow} A \rrparenthesis_R)) \mathbin{@} \llparenthesis B \stackrel{p}{\Longrightarrow} D \rrparenthesis_R \end{aligned}$$

- $\boxed{(V : A \stackrel{p}{\Longrightarrow} \forall X.B)\ X \longmapsto_{\mathsf{B}} V : A \stackrel{p}{\Longrightarrow} B}$
  We have

$$\llparenthesis A \stackrel{p}{\Longrightarrow} \forall X.B \rrparenthesis_R = \overline{\lambda} v.\, \lambda k'.\, v \mathbin{@} \llparenthesis A \stackrel{p}{\Longrightarrow} B \rrparenthesis_{R, X \mapsto (\bullet, k')}$$

  and $R(X) = (j,k)$, so

$$\begin{aligned} \llparenthesis M \rrparenthesis_R &= \llparenthesis (V : A \stackrel{p}{\Longrightarrow} \forall X.B)\ X \rrparenthesis_R \\ &= \llparenthesis V \rrparenthesis_R \mathbin{@} \llparenthesis A \stackrel{p}{\Longrightarrow} B \rrparenthesis_R\ k \\ &= (\lambda k'.\, \llparenthesis V \rrparenthesis_R \mathbin{@} \llparenthesis A \stackrel{p}{\Longrightarrow} B \rrparenthesis_{R, X \mapsto (\bullet, k')})\ k \\ &\longmapsto_{\mathsf{K}} \llparenthesis V \rrparenthesis_R \mathbin{@} \llparenthesis A \stackrel{p}{\Longrightarrow} B \rrparenthesis_{R, X \mapsto (\bullet, k')} \\ &= \llparenthesis M' \rrparenthesis_R \end{aligned}$$

- $\boxed{(V : \forall X.A \stackrel{p}{\Longrightarrow} B \longmapsto_{\mathsf{B}} (V\ \star) : A[X{:=}\star] \stackrel{p}{\Longrightarrow} B}$
  We have

$$\begin{aligned} &\llparenthesis \forall X.A \stackrel{p}{\Longrightarrow} B \rrparenthesis_R \\ &= \begin{array}{l} \overline{\lambda} v.\, \mathtt{let}\ j = \mathtt{new}\langle \mathtt{bits} \rangle\ \mathtt{in}\ \mathtt{let}\ k' = \mathtt{new}\langle \mathtt{bits} \rangle\ \mathtt{in} \\ ((v\ k') \mathbin{@} \llparenthesis A \stackrel{+X}{\Longrightarrow} A[X{:=}\star] \rrparenthesis_{R'}) \mathbin{@} \llparenthesis A[X{:=}\star] \stackrel{p}{\Longrightarrow} B \rrparenthesis_R \end{array} \end{aligned}$$

  where $R' = R, X \mapsto (j, k')$. So

$$\begin{aligned} &\llparenthesis M \rrparenthesis_R = \llparenthesis (V : \forall X.A \stackrel{p}{\Longrightarrow} B \rrparenthesis_R \\ &= \begin{array}{l} \mathtt{let}\ j = \mathtt{new}\langle \llparenthesis \star \rrparenthesis \rangle\ \mathtt{in}\ \mathtt{let}\ k' = \mathtt{new}\langle \mathtt{bits} \rangle\ \mathtt{in} \\ ((\llparenthesis V \rrparenthesis_R\ k') \mathbin{@} \llparenthesis A \stackrel{+X}{\Longrightarrow} A[X{:=}\star] \rrparenthesis_{R'} \mathbin{@} \llparenthesis A[X{:=}\star] \stackrel{p}{\Longrightarrow} B \rrparenthesis_R \end{array} \\ &= \llparenthesis \nu X{:=}\star.V\ X : A \stackrel{+X}{\Longrightarrow} A[X{:=}\star] \stackrel{p}{\Longrightarrow} B \rrparenthesis_R \\ &= \llparenthesis (V\ \star) : A[X{:=}\star] \stackrel{p}{\Longrightarrow} B \rrparenthesis_R = \llparenthesis M' \rrparenthesis_R \end{aligned}$$

- $\boxed{V : X \stackrel{p}{\Longrightarrow} X \longmapsto_{\mathsf{B}} V}$ straightforward
- $\boxed{V : \star \stackrel{p}{\Longrightarrow} \star \longmapsto_{\mathsf{B}} V}$ straightforward

- $\boxed{V : A \overset{p}{\Longrightarrow} \star \longmapsto_{\mathsf{B}} V : A \overset{p}{\Longrightarrow} G \overset{p}{\Longrightarrow} \star}$

$$\begin{aligned}\llparenthesis M \rrparenthesis_R &= \llparenthesis V : A \overset{p}{\Longrightarrow} \star \rrparenthesis_R \\ &= \llparenthesis V \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis A \overset{p}{\Longrightarrow} \star \rrparenthesis_R \\ &= \llparenthesis V \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis A \overset{p}{\Longrightarrow} G \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis G \overset{p}{\Longrightarrow} \star \rrparenthesis_R \\ &= \llparenthesis M' \rrparenthesis_R\end{aligned}$$

- $\boxed{V : \star \overset{p}{\Longrightarrow} A \longmapsto_{\mathsf{B}} V : \star \overset{p}{\Longrightarrow} G \overset{p}{\Longrightarrow} A}$

$$\begin{aligned}\llparenthesis M \rrparenthesis_R &= \llparenthesis V : \star \overset{p}{\Longrightarrow} A \rrparenthesis_R \\ &= \llparenthesis V \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis \star \overset{p}{\Longrightarrow} A \rrparenthesis_R \\ &= \llparenthesis V \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis \star \overset{p}{\Longrightarrow} G \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis G \overset{p}{\Longrightarrow} A \rrparenthesis_R \\ &= \llparenthesis M' \rrparenthesis_R\end{aligned}$$

- $\boxed{V : G \overset{p}{\Longrightarrow} \star \overset{q}{\Longrightarrow} G \longmapsto_{\mathsf{B}} V}$ We have

$$\begin{aligned}\llparenthesis G \overset{p}{\Longrightarrow} \star \rrparenthesis_R &= \overline{\lambda} v.\, \lfloor v \rfloor_k \\ \llparenthesis \star \overset{q}{\Longrightarrow} G \rrparenthesis_R &= \overline{\lambda} v.\, \lceil v \rceil_k^p\end{aligned}$$

where $k = \llparenthesis G \rrparenthesis_R$. By part 1, $\llparenthesis V \rrparenthesis_R$ is a value, so

$$\llparenthesis V : G \overset{p}{\Longrightarrow} \star \overset{q}{\Longrightarrow} G \rrparenthesis_R = \lceil \lfloor \llparenthesis V \rrparenthesis_R \rfloor_k \rceil_k^p \longmapsto_{\mathsf{K}} \llparenthesis V \rrparenthesis_R$$

- $\boxed{V : G \overset{p}{\Longrightarrow} \star \overset{q}{\Longrightarrow} H \longmapsto_{\mathsf{B}} \texttt{blame}\, q}$ We have

$$\begin{aligned}\llparenthesis G \overset{p}{\Longrightarrow} \star \rrparenthesis_R &= \overline{\lambda} v.\, \lfloor v \rfloor_k \\ \llparenthesis \star \overset{q}{\Longrightarrow} H \rrparenthesis_R &= \overline{\lambda} v.\, \lceil v \rceil_{k'}^q\end{aligned}$$

where $k = \llparenthesis G \rrparenthesis_R$ and $k' = \llparenthesis H \rrparenthesis_R$. Because $G \neq H$, we have $k \neq k'$. By part 1, $\llparenthesis V \rrparenthesis_R$ is a value, so

$$\llparenthesis V : G \overset{p}{\Longrightarrow} \star \overset{q}{\Longrightarrow} H \rrparenthesis_R = \lceil \lfloor \llparenthesis V \rrparenthesis_R \rfloor_k \rceil_{k'}^q \longmapsto_{\mathsf{K}} \texttt{blame}\, q$$

□

Proposition 17 (Simulation, $\lambda\mathsf{B}$ to $\lambda\mathsf{K}$)**.**
Assume $\Sigma \rhd M : A$ and $\Sigma \sim R$. Then

1. *If $M = V$ then $\llparenthesis M \rrparenthesis_R$ is a value.*
2. *If $\Sigma \rhd M \longrightarrow_{\mathsf{B}} \Sigma' \rhd M'$ then $\llparenthesis \Sigma \rrparenthesis_R \rhd \llparenthesis M \rrparenthesis_R \longrightarrow_{\mathsf{K}}^{*} \llparenthesis \Sigma' \rrparenthesis_{R'} \rhd \llparenthesis M' \rrparenthesis_{R'}$, for some R'.*

Proof.

1. We proceed by induction on V.
   - Case $V = c$:

     $$\llparenthesis c \rrparenthesis_R = c$$

   - Case $V = \lambda x{:}A_1.\, N[x]$:

     $$\llparenthesis \lambda x{:}A_1.\, N[x] \rrparenthesis_R = \lambda x{:}\llparenthesis A_1 \rrparenthesis.\, \llparenthesis N[X] \rrparenthesis_R$$

   - Case $V = \Lambda X.\, V'[X]$:

     $$\llparenthesis \Lambda X.\, V'[X] \rrparenthesis_R = \lambda k{:}\texttt{key}\langle\texttt{bits}\rangle.\, \llparenthesis V'[X] \rrparenthesis_{R, X \mapsto (\bullet, k)}$$

   - Case $V = V' : B_1 {\to} B_2 \overset{\phi}{\Longrightarrow} A_1 {\to} A_2$
     By induction, $\llparenthesis V' \rrparenthesis_R$ is a value. We have

     $$\begin{aligned}&\llparenthesis V' \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis B_1 {\to} B_2 \overset{\phi}{\Longrightarrow} A_1 {\to} A_2 \rrparenthesis_R \\ &= \lambda w{:}A_1.\, (\llparenthesis V' \rrparenthesis_R\ (w \mathbin{\texttt{@}} \llparenthesis A_1 \overset{-\phi}{\Longrightarrow} B_1 \rrparenthesis_{)})) \mathbin{\texttt{@}} \llparenthesis B_2 \overset{\phi}{\Longrightarrow} A_2 \rrparenthesis_R\end{aligned}$$

   - Case $V = V' : \forall X.B \overset{\phi}{\Longrightarrow} \forall X.A'$ By induction, $\llparenthesis V' \rrparenthesis_R$ is a value. We have

     $$\begin{aligned}&\llparenthesis V' \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis \forall X.B \overset{\phi}{\Longrightarrow} \forall X.A' \rrparenthesis_R \\ &= \lambda k{:}\texttt{key}\langle\texttt{bits}\rangle.\, ((\llparenthesis V' \rrparenthesis_R\ k) \mathbin{\texttt{@}} \llparenthesis B \overset{\phi}{\Longrightarrow} A' \rrparenthesis_{R, X \mapsto (\bullet, k)}\end{aligned}$$

   - Case $V = V' : B \overset{-X}{\Longrightarrow} X$
     By induction, $\llparenthesis V' \rrparenthesis_R$ is a value. We have $R(X) = (j, k)$ because $X{:=}A \in \Sigma$ and $\Sigma \sim R$, so

     $$\llparenthesis V' \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis B \overset{-X}{\Longrightarrow} X \rrparenthesis_R = \lfloor \llparenthesis V' \rrparenthesis_R \rfloor_j$$

   - Case $V = V' : B_1 {\to} B_2 \overset{p}{\Longrightarrow} A_1 {\to} A_2$: Similar to the case for $V = V' : B_1 {\to} B_2 \overset{\phi}{\Longrightarrow} A_1 {\to} A_2$.
   - Case $V = V' : B \overset{p}{\Longrightarrow} \forall X.A'$:
     By induction, $\llparenthesis V' \rrparenthesis_R$ is a value. We have

     $$\begin{aligned}&\llparenthesis V' \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis B \overset{p}{\Longrightarrow} \forall X.A' \rrparenthesis_R \\ &= \lambda k{:}\texttt{key}\langle\texttt{bits}\rangle.\, \llparenthesis V' \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis B \overset{p}{\Longrightarrow} A' \rrparenthesis_{R, X \mapsto (\bullet, K)}\end{aligned}$$

   - Case $V = V' : G \overset{p}{\Longrightarrow} \star$: By induction, $\llparenthesis V' \rrparenthesis_R$ is a value. We have

     $$\llparenthesis V' \rrparenthesis_R \mathbin{\texttt{@}} \llparenthesis G \overset{p}{\Longrightarrow} \star \rrparenthesis_R = \lfloor \llparenthesis V' \rrparenthesis_R \rfloor_{\llparenthesis G \rrparenthesis_R}$$

2. We proceed by cases on $\Sigma \rhd M \longrightarrow_{\mathsf{B}} \Sigma' \rhd M'$.
   - $\boxed{\dfrac{M \longmapsto M'}{\Sigma \vdash \mathcal{E}[M] \longrightarrow \mathcal{E}[M']}}$
     This case follows from Lemma 16.
   - $\boxed{\Sigma \rhd \mathcal{E}[\texttt{blame}\, p] \longrightarrow_{\mathsf{B}} \Sigma \rhd \texttt{blame}\, p}$
     We have $\llparenthesis \texttt{blame}\, p \rrparenthesis_R = \texttt{blame}\, p$ and

     $$\llparenthesis \Sigma \rrparenthesis_R \rhd \llparenthesis \mathcal{E} \rrparenthesis_R[\texttt{blame}\, p] \longrightarrow_{\mathsf{K}} \llparenthesis \Sigma \rrparenthesis_R \rhd \texttt{blame}\, p$$

   - $\boxed{\Sigma \rhd \mathcal{E}[\nu X{:=}A.N] \longrightarrow_{\mathsf{B}} \Sigma, X{:=}A \rhd \mathcal{E}[N]}$
     Pick $R' = R, X \mapsto (\kappa_j, \kappa_k)$. We have

     $$\begin{aligned}&\llparenthesis \Sigma \rrparenthesis_R \rhd \llparenthesis \mathcal{E} \rrparenthesis_R[\llparenthesis \nu X{:=}A.N \rrparenthesis_R] \\ &\longrightarrow_{\mathsf{K}}^{*} \llparenthesis \Sigma, X{:=}A \rrparenthesis_{R'} \rhd \llparenthesis \mathcal{E} \rrparenthesis_R[\llparenthesis N \rrparenthesis_{R'}]\end{aligned}$$

     because

     $$\llparenthesis \nu X{:=}A.N \rrparenthesis_R = \begin{array}{l}\texttt{let}\ j = \texttt{new}\langle\llparenthesis A \rrparenthesis\rangle\ \texttt{in} \\ \texttt{let}\ k = \texttt{new}\langle\texttt{bits}\rangle\ \texttt{in} \\ \llparenthesis N \rrparenthesis_{R, X \mapsto (j,k)}\end{array}$$

     and

     $$\llparenthesis \Sigma, X{:=}A \rrparenthesis_{R'} = \llparenthesis \Sigma \rrparenthesis, \kappa_j : \texttt{key}\langle\llparenthesis A \rrparenthesis\rangle, \kappa_k : \texttt{key}\langle\texttt{bits}\rangle$$

□

6. Results for Translation between λF and λG

Compilation and decompilation preserve values.

Lemma 18 (Preservation of values)**.**

- *$\llparenthesis V \rrparenthesis = V'$, for some value V'.*
- *$\llparenthesis V' \rrparenthesis = V$, for some value V.*

Decompilation is the inverse of compilation.

Lemma 19 (Compiling and decompiling)**.** $\llparenthesis \llparenthesis M \rrparenthesis \rrparenthesis = M$.

6.1 Proof of Proposition 12 of the paper

Decompilation is a bisimulation.

Proposition 20 (Decompilation is a bisimulation)**.**
Assume $\Sigma \vdash_{\lambda\mathsf{G}} M' : A$ and $\llparenthesis \Sigma\ ;\ M' \rrparenthesis = M$.

- *If* $\Sigma\,;M' \longrightarrow_{\lambda\mathsf{G}} \Pi\,;N'$ *then* $M \longrightarrow^{?}_{\lambda} N$ *and* $\llparenthesis \Pi\,;N' \rrparenthesis = N$ *for some* N.
- *If* $M \longrightarrow_{\lambda\mathsf{F}} N$ *then* $\Sigma\,;M' \longrightarrow^{*}_{\lambda\mathsf{F}} \Pi\,;N'$ *and* $\llparenthesis \Pi\,;N' \rrparenthesis = N$ *for some* Π, N'.

The first part is proved by case analysis over reductions of $\lambda\mathsf{G}$. The second part is proved using the following lemmas.

Lemma 21. *Assume* $\Sigma \vdash_{\lambda\mathsf{G}} M' : A$ *and* $\llparenthesis \Sigma\,;M \rrparenthesis = op(\vec{V})$. *Then either:*

- $M' = op(\vec{V'})$ *and* $\llparenthesis \Sigma\,;\vec{V'} \rrparenthesis = \vec{V}$ *for some* $\vec{V'}$, *or*
- $\Sigma\,;M' \longrightarrow_{\lambda\mathsf{G}} \Pi\,;N'$ *and* $\llparenthesis \Sigma\,;M' \rrparenthesis = \llparenthesis \Pi\,;N' \rrparenthesis$ *for some* Π, N'.

Lemma 22. *Assume* $\Sigma \vdash_{\lambda\mathsf{G}} M' : B$ *and* $\llparenthesis \Sigma\,;M' \rrparenthesis = (\lambda x.\, N)\ V$. *Then either:*

- $M' = (\lambda x{:}A.\, N')V'$ *and* $\llparenthesis \Sigma\,;N' \rrparenthesis = N$, $\llparenthesis \Sigma\,;V' \rrparenthesis = V$ *for some* A, N', V', *or*
- $\Sigma\,;M' \longrightarrow_{\lambda\mathsf{G}} \Pi\,;N'$ *and* $\llparenthesis \Sigma\,;M' \rrparenthesis = \llparenthesis \Pi\,;N' \rrparenthesis$ *for some* Π, N'.

Lemma 23. *Assume* $\Sigma \vdash_{\lambda\mathsf{G}} M' : B$ *and* $\llparenthesis \Sigma;M' \rrparenthesis = (\Lambda X.\, N)\ C$. *Then either:*

- $M' = (\Lambda X.\, N')C$ *and* $\llparenthesis \Sigma\,;N' \rrparenthesis = N$ *for some* N', *or*
- $\Sigma\,;M' \longrightarrow_{\lambda\mathsf{G}} \Pi\,;N'$ *and* $\llparenthesis \Sigma\,;M' \rrparenthesis = \llparenthesis \Pi\,;N' \rrparenthesis$ *for some* Π, N'.

We consider the proof of the third lemma, the other two are similar. By progress, the term M' must reduce. We cannot have reduction of an operator, since then the erasure would not be as given. The only possible reductions are reduction of a type application, in which case the first clause holds, or reduction of a binding or of a static cast, in which case the second clause holds.

We have the following easy corollary of bisimulation.

Corollary 24. *Assume* $\cdot \vdash_{\lambda\mathsf{G}} M' : \iota$. *Then* $M' \longrightarrow^{*}_{\lambda\mathsf{G}} k$ *iff* $\llparenthesis M' \rrparenthesis \longrightarrow^{*}_{\lambda\mathsf{F}} k$.

6.2 Proof of Proposition 13 of the paper

Let $\mathcal{C}$ range over arbitrary contexts of Λ, and $\mathcal{C}'$ range over arbitrary contexts of $\lambda\mathsf{G}$.

Definition 25 (Context typing). *For a context* $\mathcal{C}$ *of* Λ, *write*

$$\mathcal{C} : (\Gamma \vdash A) \Longrightarrow (\Delta \vdash B)$$

if for any M *such that* $\Gamma \vdash M : A$ *it holds that* $\Delta \vdash \mathcal{C}[M] : B$.

The same holds, mutatis mutandis, for $\lambda\mathsf{G}$.

It is straightforward to give direct rules for typing contexts, in the style of Ahmed and Blume (2011), but the above definition suffices for our purposes.

We extend compiling and decompiling to contexts in the obvious way. Both compilation and decompilation preserve substitution.

Lemma 26 (Substitution)**.**

- $\llparenthesis \mathcal{C}[M] \rrparenthesis = \llparenthesis \mathcal{C} \rrparenthesis[\llparenthesis M \rrparenthesis]$.
- $\llparenthesis \mathcal{C}'[M'] \rrparenthesis = \llparenthesis \mathcal{C}' \rrparenthesis[\llparenthesis M' \rrparenthesis]$.

We define contextual equivalence as usual.

Definition 27 (Contextual equivalence)**.** *Suppose* $\Gamma \vdash M : A$ *and* $\Gamma \vdash N : A$. *We say* M *and* N *are* contextually equivalent*, and write* $M =_{\lambda\mathsf{F}} N$, *if for every context* $\mathcal{C} : (\Gamma \vdash A) \Longrightarrow (\cdot \vdash \iota)$ *we have* $\mathcal{C}[M] \longrightarrow^{*}_{\lambda\mathsf{F}} k$ *if and only if* $\mathcal{C}[N] \longrightarrow^{*}_{\lambda\mathsf{F}} k$.

The same applies, mutatis mutandis, for $\lambda\mathsf{G}$.

Decompilation is fully abstract.

Proposition 28 (Decompilation is fully abstract)**.** *Assume* $\Gamma' \vdash M' : A'$ *and* $\Gamma' \vdash N' : A'$. *Then* $M' =_{\lambda\mathsf{G}} N'$ *iff* $\llparenthesis M' \rrparenthesis =_{\lambda\mathsf{F}} \llparenthesis N' \rrparenthesis$.

In the forward direction, we have for all $\mathcal{C}'$ that $\mathcal{C}'[M'] \longrightarrow^{*}_{\lambda\mathsf{G}} k$ iff $\mathcal{C}'[N'] \longrightarrow^{*}_{\lambda\mathsf{G}} k$, and we need to show for all $\mathcal{C}$ that $\mathcal{C}[\llparenthesis M \rrparenthesis] \longrightarrow^{*}_{\lambda\mathsf{F}} k$ iff $\mathcal{C}[N] \longrightarrow^{*}_{\lambda\mathsf{F}} k$. Given $\mathcal{C}$, take $\mathcal{C}' = \llparenthesis CC \rrparenthesis$. Then $\llparenthesis \mathcal{C}'[M'] \rrparenthesis = \mathcal{C}[\llparenthesis M' \rrparenthesis]$, by Lemmas 26 and 19, and similarly for N and N'. The result then follows by Corollary 24.

In the backward direction, we have for all $\mathcal{C}$ that $\mathcal{C}[\llparenthesis M \rrparenthesis] \longrightarrow^{*}_{\lambda\mathsf{F}} k$ iff $\mathcal{C}[N] \longrightarrow^{*}_{\lambda\mathsf{F}} k$ and we need to show for all $\mathcal{C}'$ that $\mathcal{C}'[M'] \longrightarrow^{*}_{\lambda\mathsf{G}} k$ iff $\mathcal{C}'[N'] \longrightarrow^{*}_{\lambda\mathsf{G}} k$. Given $\mathcal{C}'$, take $\mathcal{C} = \llparenthesis \mathcal{C}' \rrparenthesis$. Then $\llparenthesis \mathcal{C}'[M'] \rrparenthesis = \mathcal{C}\llparenthesis M \rrparenthesis$, and similarly for N and N'. The results then follows by Corollary 24.

As corollary, we have that compilation is fully abstract.

Proposition 29 (Compilation is fully abstract)**.** *Assume* $\Gamma \vdash M : A$ *and* $\Gamma \vdash N : A$. *Then* $M =_{\lambda\mathsf{F}} N$ *iff* $\llparenthesis M \rrparenthesis =_{\lambda\mathsf{G}} \llparenthesis N \rrparenthesis$.

This follows immediately from Proposition 28. Take $M' = \llparenthesis M \rrparenthesis$ and $N' = \llparenthesis N \rrparenthesis$ and apply Lemma 19.

References

A. Ahmed and M. Blume. An equivalence-preserving CPS translation via multi-language semantics. In *International Conference on Functional Programming (ICFP)*, pages 431–444, 2011.