Witnessing Mutability, Purity and Aliasing for Program Optimisation (Appendix)

BEN LIPPMEIER
School of Computer Science and Engineering
University of New South Wales, Australia
(*e-mail:* benl@ouroborus.net)

1 Formalities

Definition: (Substitution)

We use standard capture avoiding substitution over terms, types, kinds, superkinds and type environments. The expression $A[B/x]$ means “A with B substituted for x”. We work up to renaming of bound variables.

Definition: (Store Model $\Sigma \models H$)

Given a store typing Σ and a store H, the store typing *models* the store, written $\Sigma \models H$ when all elements of the store typing correspond to elements of the store.

$$\begin{array}{lll}
\text{for all} & l :: T\ \underline{\rho}\ \overline{\varphi} \in \Sigma & \text{we have}\ \ l \overset{\rho}{\mapsto} C_K\ \overline{v^\circ} \in H \\
\text{for all} & \underline{\rho} \in \Sigma & \text{we have}\ \ \underline{\rho} \in H \\
\text{for all} & \underline{\text{mutable}\ \rho} \in \Sigma & \text{we have}\ \ \text{mutable}\ \rho \in H \\
\text{for all} & \underline{\text{const}\ \rho} \in \Sigma & \text{we have}\ \ \text{const}\ \rho \in H
\end{array}$$

Definition: (Well Typed Stores)

Given a store typing Σ and a store H, the store is *well typed* relative to the store typing, written $\Sigma \vdash H$ when all elements of the store correspond to elements in the store typing.

$$\begin{array}{ll}
\text{for all} & l \overset{\rho}{\mapsto} C_K\ \overline{v^\circ} \in H \\
\ \ \text{we have} & l :: T\ \underline{\rho}\ \overline{\varphi} \in \Sigma \\
\ \ \text{and} & K :: \forall (r : \%) (\overline{a : \kappa}) . \overline{\tau} \to T\ r\ \overline{a} \in \text{ctorTypes}(T) \\
\ \ \text{and} & \overline{\emptyset \mid \Sigma \vdash v_i^\circ\ ::\ \tau_i[\underline{\rho}/r\ \overline{\varphi/a}]\ ;\ \bot}^{\,i}
\end{array}$$

$$\begin{array}{lll}
\text{for all} & \underline{\rho} \in H & \text{we have}\ \ \underline{\rho} \in \Sigma \\
\text{for all} & \text{mutable}\ \rho \in H & \text{we have}\ \ \underline{\text{mutable}\ \rho} \in \Sigma \\
\text{for all} & \text{const}\ \rho \in H & \text{we have}\ \ \underline{\text{const}\ \rho} \in \Sigma
\end{array}$$

Definition: (No Fabricated Region Witnesses)

The predicate nofab() is short for "No fabricated region witnesses" and refers to the syntactic restriction that the witness constructors *MkMutable* and *MkConst* may only appear in the witness list associated with a **letregion**. The predicate is defined for value, type, kind and superkind expressions, as all can contain type applications.

Example inference rules are as follows:

$$\frac{\mathrm{nofab}(t) \quad \mathrm{nofab}(\varphi)}{\mathrm{nofab}(t\ \varphi)}$$

$$\frac{\mathrm{nofab}(\varphi_1) \quad \mathrm{nofab}(\varphi_2)}{\mathrm{nofab}(\varphi_1\ \varphi_2)}$$

$$\frac{\mathrm{nofab}(t)}{\mathrm{nofab}(\textbf{letregion}\ r\ \textbf{with}\ \overline{w = \delta}\ \textbf{in}\ t)}$$

Besides this last judgement which ignores the witness list associated with a **letregion**, all others are generated by mechanical recursive decent of the syntax tree. Clearly, nofab($MkMutable$) and nofab($MkConst$) must be *false*.

2 Proofs of Language Properties

Here we present the formal proofs of language properties. Each of these proofs is by induction over the derivation of a typing judgement. When presenting each case, we assume the statement being considered and then invoke the standard inversion lemmas to fill in the appropriate premises.

For example, the proof of Substitution of Values in Values contains the case $\Lambda(a : \kappa).\, t_1$. We assume the statement $\Gamma,\ x : \tau_2 \,|\, \Sigma \vdash \Lambda(a : \kappa).t_1 \ ::\ \forall(a : \kappa).\, \tau_1\ ;\ \sigma$, and use the inversion lemma to give $\Gamma,\ x : \tau_2,\ a : \kappa \,|\, \Sigma \vdash t_1\ ::\ \tau_1\ ;\ \sigma$

$$\frac{(4)\ \Gamma,\ x : \tau_2,\ a : \kappa \,|\, \Sigma \vdash t_1\ ::\ \tau_1\ ;\ \sigma}{(\underline{1})\ \Gamma,\ x : \tau_2 \,|\, \Sigma \vdash \Lambda(a : \kappa).t_1\ ::\ \forall(a : \kappa).\, \tau_1\ ;\ \sigma}$$

Each statement is numbered for identification purposes, and we underline the numbers of statements that are assumptions. In some cases not all statements obtained by the inversion lemmas will be used, but we include them as premises anyway so that the typing rules maintain their familiar shapes.

We will omit the quantifiers "for all" and "for some" when they are obvious from the context, as they clutter the proof without providing much additional information.

Lemma: (Forms of Terms and Types)
When a term is in normal form we can determine its shape by inspecting its type. Similarly, when a type is in normal form we can determine its shape by inspecting its kind. For example:

$$\begin{array}{rl} \text{If} & t \in \mathit{Value} \\ \text{and} & \emptyset \,|\, \Sigma \vdash t \,::\, \tau_1 \to \tau_2 \,;\, \sigma \\ \text{then} & t = \lambda(x : \tau_1).\, t' \end{array}$$

By inspection of the typing rules. The only values that can have function types are lambda abstractions and variables, but as the type environment is empty the value cannot be a variable. Note that locations cannot have function types because abstractions in the store always appear inside data constructors.

Lemma: (No free witness variables in effects)

$$\begin{array}{rl} \text{If} & \Gamma \,|\, \Sigma \vdash t \,::\, \varphi \,;\, \sigma \\ \text{and} & \Gamma \,|\, \Sigma \vdash_{\text{T}} w \,::\, \kappa \text{ and } \Gamma \,|\, \Sigma \vdash_{\text{K}} \kappa \,::\, \Diamond \\ \text{then} & \sigma[\delta/w] \equiv \sigma \end{array}$$

By inspection of the kinding rules for effect constructors.

Lemma: (No free witness variables in term types)

$$\begin{array}{rl} \text{If} & \Gamma \,|\, \Sigma \vdash t \,::\, \varphi \,;\, \sigma \\ \text{and} & \Gamma \,|\, \Sigma \vdash_{\text{T}} w \,::\, \kappa \text{ and } \Gamma \,|\, \Sigma \vdash_{\text{K}} \kappa \,::\, \Diamond \\ \text{then} & \varphi[\delta/w] \equiv \varphi \end{array}$$

By inspection of the kinding rules for value type constructors.

Lemma: (Weaken Store Typing)

If we can assign a term t some type and effect, then we can also assign t the same type and effect under a larger store typing. This property is also true for kind judgements.

$$\begin{array}{rl} \text{If} & \Gamma \,|\, \Sigma \vdash t \,::\, \tau \,;\, \sigma \\ \text{and} & \Sigma' \supseteq \Sigma \\ \text{then} & \Gamma \,|\, \Sigma' \vdash t \,::\, \tau \,;\, \sigma \end{array}$$

By induction over the derivation of $\Gamma \,|\, \Sigma \vdash t \,::\, \tau \,;\, \sigma$. At the top of the derivation tree we will have uses of (TyLoc) that include statements such as $\Gamma \,|\, \Sigma,\, l : \tau \vdash l \,::\, \tau \,;\, \bot$. These statements remain true when Σ is extended.

Lemma: (Strengthen Type Environment)

$$\begin{array}{rl} \text{If} & \Gamma,\, x : \tau \,|\, \Sigma \vdash_{\text{T}} \varphi \,::\, \kappa \\ \text{then} & \Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi \,::\, \kappa \end{array}$$

By inspection of the forms of types. Types to not contain value variables.

Lemma: (Weaken Type Environment)

$$\begin{array}{ll} \text{If} \quad \Gamma \,|\, \Sigma \vdash t \,::\, \tau_1 \,;\, \sigma & \text{If} \quad \Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi \,::\, \kappa \\ \text{and} \quad x \notin fv(t) & \text{and} \quad a \notin fv(\varphi) \\ \text{then} \quad \Gamma,\, x : \tau_2 \,|\, \Sigma \vdash t \,::\, \tau_1 \,;\, \sigma & \text{then} \quad \Gamma,\, a : \varphi_2 \,|\, \Sigma \vdash_{\text{T}} \varphi \,::\, \kappa \end{array}$$

By induction over the derivations of the first statements of each.

Lemma: (Region Witness Assertion)

If we add a property to the heap, then we can always evaluate the witness constructor that tests for it. The following statement is true

$$H,\ \text{propOf}(\Delta)\ ;\ \delta \rightsquigarrow \Delta \text{ and } \delta \in \{MkConst\ r, MkMutable\ r\} \text{ for some } r,\, \Delta.$$

By inspection of the transition rules (EwConst) and (EwMutable).

Lemma: (Progress of Purity)

$$\begin{array}{ll} \text{If} & \emptyset \,|\, \Sigma \vdash_{\text{T}} \delta \,::\, Pure\ \sigma \\ \text{and} & \text{nofab}(\delta) \\ \text{then} & \delta = \underline{\text{pure } \sigma} \\ \text{or} & H ; \delta \rightsquigarrow \delta' \text{ for all } H \text{ and some } \delta'. \end{array}$$

Proof: By induction over the derivation of $\emptyset \,|\, \Sigma \vdash_{\text{T}} \delta \,::\, Pure\ \sigma$

(IH)	Progress of Purity holds for all subterms of δ.	(assume)
($\underline{1}$)	$\emptyset \,|\, \Gamma \vdash_{\text{T}} \delta \,::\, Pure\ \sigma$	(assume)
($\underline{2}$)	$\text{nofab}(\delta)$	(assume)
(3)	$\delta \in \{\underline{\text{pure } \sigma},\ MkPureBot,\ MkPurify\ \underline{\rho}\ \delta_1,$ $MkPureJoin\ \sigma_2\ \sigma_3\ \delta_2\ \delta_3\}$	(Forms of Types 1)
Case:	$\delta = \underline{\text{pure } \sigma}$	
(4)	immediate	
Case:	$\delta = MkPureBot$	
(5)	$H\ ;\ MkPureBot \rightsquigarrow \underline{\text{pure } \bot}$	(EwPureBot)
Case:	$\delta = MkPurify\ \underline{\rho}\ \delta_1$	
(6)	$\delta_1 = \underline{\text{const } \rho}$	(Kind of $MkPurify$ 2)
(7)	$H\ ;\ MkPurify\ \underline{\rho}\ \underline{\text{const } \rho} \rightsquigarrow \underline{\text{pure } (Read\ \rho)}$	(EwPurify 6)
Case:	$\delta = MkPureJoin\ \sigma_2\ \sigma_3\ \delta_2\ \delta_3$	
(8)	$\emptyset \,|\, \Sigma \vdash_{\text{T}} \delta_2 \,::\, Pure\ \sigma_2$	(Kind of $MkPureJoin$ 1)
(9)	$\text{nofab}(\delta_2)$	(Def. nofab 2)
(10)	$\delta_2 = \underline{\text{pure } \sigma_2}$ or $H ; \delta_2 \rightsquigarrow \delta_2'$	(IH 8 9)
(11)	$\delta_3 = \underline{\text{pure } \sigma_3}$ or $H ; \delta_3 \rightsquigarrow \delta_3'$	(Similarly)
(12)	Either (EwContext) or (EwPureJoin) applies.	

Lemma: (Preservation of Purity)

$$\begin{array}{rl} \text{If} & \emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \delta :: \mathit{Pure}\ \sigma \\ \text{and} & H\,;\,\delta \rightsquigarrow \delta' \\ \text{then} & \emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \delta' :: \mathit{Pure}\ \sigma \end{array}$$

Proof: By induction over the derivation of $\emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \delta :: \mathit{Pure}\ \sigma$

(IH) Preservation of Purity holds for all subterms of δ. (assume)

Case: $\delta = \mathit{MkPureBot}$ / EwPureBot

($\underline{1}$)	$\emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \mathit{MkPureBot} :: \mathit{Pure}\ \bot$	(assume)
($\underline{2}$)	$H\,;\,\mathit{MkPureBot} \rightsquigarrow \underline{\mathrm{pure}\ \bot}$	(assume)
(3)	$\emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \underline{\mathrm{pure}\ \bot} :: \mathit{Pure}\ \bot$	(KiPure)

Case: $\delta = \mathit{MkPurify}\ \underline{\rho}\ \underline{\mathrm{const}\ \rho}$ / EwPurify

$$\dfrac{\dfrac{\vdots}{(3)\ \emptyset \,|\, \Sigma \vdash_{\mathrm{T}} (\mathit{MkPurify}\ \underline{\rho}) :: \mathit{Const}\ \underline{\rho} \to \mathit{Pure}\ (\mathit{Read}\ \underline{\rho})} \quad (4)\ \emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \underline{\mathrm{const}\ \rho} :: \mathit{Const}\ \underline{\rho}}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \mathit{MkPurify}\ \underline{\rho}\ \underline{\mathrm{const}\ \rho} :: \mathit{Pure}\ (\mathit{Read}\ \underline{\rho})}$$

$$(\underline{2})\ H\,;\,\mathit{MkPurify}\ \underline{\rho}\ \underline{\mathrm{const}\ \rho} \rightsquigarrow \underline{\mathit{pure}\ (\mathit{Read}\ \rho)}$$

(5)	$\underline{\mathrm{const}\ \rho} \in \Sigma$	(KiConst 4)
(6)	$\emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \underline{\mathrm{pure}\ (\mathit{Read}\ \rho)} :: \mathit{Pure}\ (\mathit{Read}\ \underline{\rho})$	(KiPurify 6)

The remaining cases are similar to the EwPurify case.

Lemma: (Substitution of Values in Values)

$$\begin{array}{rl} \text{If} & \Gamma,\, x : \tau_2 \,|\, \Sigma \vdash t \,::\, \tau_1 \,;\, \sigma \\ \text{and} & \Gamma \,|\, \Sigma \vdash v^\circ \,::\, \tau_2 \,;\, \bot \\ \text{then} & \Gamma \,|\, \Sigma \vdash t[v^\circ/x] \,::\, \tau_1 \,;\, \sigma \end{array}$$

Proof: By induction over the derivation of $\Gamma,\, x : \tau_2 \,|\, \Sigma \vdash t \,::\, \tau_1 \,;\, \sigma$

(IH)	Subst. of Values in Values holds for all subterms of t.	(assume)

Case: $t = y$ / TyVar
Trivial. $x \neq y$ so t is unaffected.

Case: $t = x$ / TyVar

$$(\underline{1})\ \Gamma,\, x : \tau_1,\, x : \tau_2 \,|\, \Sigma \vdash x \,::\, \tau_1 \,;\, \bot$$

($\underline{2}$)	$\Gamma \,\vert\, \Sigma \vdash v^\circ \,::\, \tau_2 \,;\, \bot$	(assume)
(3)	$\tau_1 \equiv \tau_2$	(Def. Type Env 1)
(4)	$\Gamma \,\vert\, \Sigma \vdash x[v^\circ/x] \,::\, \tau_1 \,;\, \bot$	(Def. Sub. 2 3)

Case: $t = l$ / TyLoc
Trivial. Locations do not contain value variables.

Case: $t = ()$ / TyUnit
Trivial. Unit does not contain value variables.

Case: $t = \Lambda(a : \kappa).\, t_1$ / TyAbsT

$$\frac{(3)\ \Gamma,\, x : \tau_2,\, a : \kappa \,|\, \Sigma \vdash t_1 \,::\, \tau_1 \,;\, \sigma}{(\underline{1})\ \Gamma,\, x : \tau_2 \,|\, \Sigma \vdash \Lambda(a : \kappa).\, t_1 \,::\, \forall(a : \kappa).\, \tau_1 \,;\, \sigma}$$

($\underline{2}$)	$\Gamma \,\vert\, \Sigma \vdash v^\circ \,::\, \tau_2 \,;\, \bot$	(assume)
(4)	$\Gamma,\, a : \kappa \,\vert\, \Sigma \vdash v^\circ \,::\, \tau_2 \,;\, \bot$	(Weak. Type Env 2)
(5)	$\Gamma,\, a : \kappa \,\vert\, \Sigma \vdash t_1[v^\circ/x] \,::\, \tau_1 \,;\, \sigma$	(IH 3 4)
(6)	$\Gamma \,\vert\, \Sigma \vdash \Lambda(a : \kappa).\, (t_1[v^\circ/x]) \,::\, \tau_1 \,;\, \sigma$	(TyAbsT 5)
(7)	$\Gamma \,\vert\, \Sigma \vdash (\Lambda(a : \kappa).\, t_1)[v^\circ/x] \,::\, \tau_1 \,;\, \sigma$	(Def. Sub. 6)

Case: $t = \lambda(x : \tau).\, t_1$ / TyAbs
Similarly to TyAbsT case.

Case: $t = t_1\ \varphi_2$ / TyAppT

$$\frac{(3)\ \Gamma,\, x : \tau_2 \,|\, \Sigma \vdash t_1 \,::\, \forall(a : \kappa_{11}).\, \varphi_{12} \,;\, \sigma \quad (4)\ \Gamma,\, x : \tau_2 \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 \,::\, \kappa_{11}}{(\underline{1})\ \Gamma,\, x : \tau_2 \,|\, \Sigma \vdash t_1\ \varphi_2 \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma[\varphi_2/a]}$$

($\underline{2}$)	$\Gamma \,\vert\, \Sigma \vdash v^\circ \,::\, \tau_2 \,;\, \bot$	(assume)
(5)	$\Gamma \,\vert\, \Sigma \vdash t_1[v^\circ/x] \,::\, \forall(a : \kappa_{11}).\, \varphi_{12} \,;\, \sigma$	(IH 3 2)
(6)	$\Gamma \,\vert\, \Sigma \vdash_{\mathrm{T}} \varphi_2 \,::\, \kappa_2$	(Str. Type Env 4)
(7)	$\Gamma \,\vert\, \Sigma \vdash t_1[v^\circ/x]\ \varphi_2 \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma[\varphi_2/a]$	(TyAppT 5 6)
(8)	$\Gamma \,\vert\, \Sigma \vdash (t_1\ \varphi_2)[v^\circ/x] \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma[\varphi_2/a]$	(Def. Sub. 7)

Case: $t = t_1\ t_2$ / TyApp

$$\frac{(3)\ \Gamma,\ x : \tau_3 \,|\, \Sigma \vdash t_1 \ ::\ \tau_{11} \overset{\sigma}{\to} \tau_{12}\,;\sigma_1 \qquad (4)\ \Gamma,\ x : \tau_3 \,|\, \Sigma \vdash t_2 \ ::\ \tau_2\,;\sigma_2}{(\underline{1})\ \Gamma,\ x : \tau_3 \,|\, \Sigma \vdash t_1\ t_2 \ ::\ \tau_{12}\,;\sigma_1 \vee \sigma_2 \vee \sigma}$$

$(\underline{2})$	$\Gamma \,|\, \Sigma \vdash v^\circ \ ::\ \tau_3\,;\bot$	(assume)
(5)	$\Gamma \,|\, \Sigma \vdash t_1[v^\circ/x] \ ::\ \tau_{11} \overset{\sigma}{\to} \tau_{12}\,;\sigma_1$	(IH 3 2)
(6)	$\Gamma \,|\, \Sigma \vdash t_2[v^\circ/x] \ ::\ \tau_2\,;\sigma_2$	(IH 4 2)
(7)	$\Gamma \,|\, \Sigma \vdash t_1[v^\circ/x]\ t_2[v^\circ/x] \ ::\ \tau_{12}\,;\sigma_1 \vee \sigma_2 \vee \sigma$	(TyApp 5 6)
(8)	$\Gamma \,|\, \Sigma \vdash (t_1\ t_2)[v^\circ/x] \ ::\ \tau_{12}\,;\sigma_1 \vee \sigma_2 \vee \sigma$	(Def. Sub. 7)

Case: $t = (\textbf{case}\ t\ \textbf{of}\ \overline{K\ \overline{x : \tau} \to t'})$ / TyCase
Similarly to TyApp case. Uses the following lemma.

$$\begin{array}{rl}
\text{If} & \Gamma,\ x : \tau_2 \,|\, \Sigma \vdash K\ \overline{y} \to t \ ::\ T\ \varphi\ \overline{\varphi'} \to \tau_1\,;\sigma \\
\text{and} & \Gamma \,|\, \Sigma \vdash v^\circ \ ::\ \tau_2\,;\bot \\
\text{then} & \Gamma \,|\, \Sigma \vdash K\ \overline{y} \to t[v^\circ/x] \ ::\ T\ \varphi\ \overline{\varphi'} \to \tau_1\,;\sigma
\end{array}$$

Case: $t = (\textbf{mask}\ \delta\ \textbf{in}\ t)$ / TyMask
Case: $t = (\textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i :: \delta_i}\}\ \textbf{in}\ t_1)$ / TyLetRegion
Case: $t = K\ \overline{\varphi}\ \overline{t}$ / ...
Case: $t = \mathit{update}_{K,i}\ \overline{\varphi}\ t_1\ t_2$ / TyUpdate
Case: $t = \mathit{suspend}\ \overline{\varphi}\ t_1\ t_2$ / TySuspend
Similarly to TyApp case. Note that value variables do not appear in type (or witness) expressions.

Lemma: (Substitution of Types in Values)

$$\begin{array}{rl} \text{If} & \Gamma,\, a : \kappa_2 \,|\, \Sigma \vdash t \,::\, \tau_1 \,;\, \sigma \\ \text{and} & \Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \,::\, \kappa_2 \\ \text{then} & \Gamma[\varphi_2/a] \,|\, \Sigma \vdash t[\varphi_2/a] \,::\, \tau_1[\varphi_2/a] \,;\, \sigma[\varphi_2/a] \end{array}$$

Proof: By induction over the derivation of $\Gamma,\, a : \kappa_2 \,|\, \Sigma \vdash t \,::\, \tau_1 \,;\, \sigma$

(IH) Subst. of Types in Values holds for all subterms of t. (assume)

Case: $t = x$ / TyVar
Case: $t = ()$ / TyUnit
Case: $t = l$ / TyLoc
Trivial. No free type vars.

Case: $t = \Lambda(a : \kappa).\, t$ / TyAbsT

$$\frac{(3)\ \Gamma,\, a : \kappa_2,\, a_{11} : \kappa_{11} \,|\, \Sigma \vdash t_{12} \,::\, \tau_{12} \,;\, \sigma_1}{(\underline{1})\ \Gamma,\, a : \kappa_2 \,|\, \Sigma \vdash \Lambda(a_{11} : \kappa_{11}).\, t_{12} \,::\, \forall(a_{11} : \kappa_{11}).\, \tau_{12} \,;\, \sigma_1}$$

($\underline{2}$)	$\Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \,::\, \kappa_2$	(assume)
(4)	$\Gamma,\, a_{11} : \kappa_{11} \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \,::\, \kappa_2$	(Weak. Type Env 2)
(5)	$(\Gamma,\, a_{11} : \kappa_{11})[\varphi_2/a] \,|\, \Sigma$ $\vdash t_{12}[\varphi_2/a] \,::\, \tau_{12}[\varphi_2/a] \,;\, \sigma_1[\varphi_2/a]$	(IH 3 4)
(6)	$\Gamma[\varphi_2/a],\, a_{11} : \kappa_{11}[\varphi_2/a] \,|\, \Sigma$ $\vdash t_{12}[\varphi_2/a] \,::\, \tau_{12}[\varphi_2/a] \,;\, \sigma_1[\varphi_2/a]$	(Def. Sub. 5)
(7)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash (\Lambda(a_{11} : \kappa_{11}).\, t_{12})[\varphi_2/a]$ $::\, (\forall(a_{11} : \kappa_{11}).\, \tau_{12})[\varphi_2/a] \,;\, \sigma_1[\varphi_2/a]$	(TyAbsT 6, Def. Sub)

Case: $t = \lambda(x : \tau_{11}).\, t_{12}$ / TyAbs
Similarly to TyAbsT case

Case: $t = t_{11}\ \varphi_{12}$ / TyAppT

$$\frac{(3)\ \Gamma,\, a : \kappa_3 \,|\, \Sigma \vdash t_1 \,::\, \forall(a_1 : \kappa_{11}).\, \varphi_{12} \,;\, \sigma_1 \quad (4)\ \Gamma,\, a : \kappa_3 \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \,::\, \kappa_{11}}{(\underline{1})\ \Gamma,\, a : \kappa_3 \,|\, \Sigma \vdash t_1\ \varphi_2 \,::\, \varphi_{12}[\varphi_2/a_1] \,;\, \sigma_1[\varphi_2/a_1]}$$

($\underline{2}$)	$\Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi_3 \,::\, \kappa_3$	(assume)
(5)	$\Gamma[\varphi_3/a] \,|\, \Sigma$ $\vdash t_1[\varphi_3/a] \,::\, (\forall(a_1 : \kappa_{11}).\, \varphi_{12})[\varphi_3/a] \,;\, \sigma_1[\varphi_3/a]$	(IH 3 2)
(6)	$\Gamma[\varphi_3/a] \,|\, \Sigma$ $\vdash t_1[\varphi_3/a] \,::\, \forall(a_1 : \kappa_{11}[\varphi_3/a]).\, \varphi_{12}[\varphi_3/a] \,;\, \sigma_1[\varphi_3/a]$	(Def. Sub. 5)
(7)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash_{\text{T}} \varphi_2[\varphi_3/a] \,::\, \kappa_{11}[\varphi_3/a]$	(Sub. Type/Type 4 2)
(8)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash t_1[\varphi_3/a]\ \ \varphi_2[\varphi_3/a]$ $::\, (\varphi_{12}[\varphi_3/a])[\varphi_2[\varphi_3/a]/a_1];\ (\sigma_1[\varphi_3/a])[\varphi_2[\varphi_3/a]/a_1]$	(TyAppT 6 7)
(9)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash (t_1\ \varphi_2)[\varphi_3/a]$ $::\, (\varphi_{12}[\varphi_2/a_1])[\varphi_3/a];\ (\sigma_1[\varphi_2/a_1])[\varphi_3/a]$	(Def. Sub. 8)

Case: $t = (t_1\ t_2)$ / TyApp
Similarly to TyAbsT case

Case: $t = \textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i = \delta_i}\}\ \textbf{in}\ t_1$ / TyLetRegion

$$\frac{\begin{array}{ll}(3)\ \overline{\delta_i}\ \text{well formed} & (5)\ \overline{\Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash_{\mathrm{T}} \kappa_i \,::\, \Diamond}^{\,i} \\ (4)\ \Gamma,\ a : \kappa_2,\ r : \%,\ \overline{w_i = \kappa_i} \,|\, \Sigma \vdash t_1 \,::\, \tau\,;\sigma & (6)\ \overline{\Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash_{\mathrm{T}} \delta_i \,::\, \kappa_i}^{\,i}\end{array}}{(\underline{1})\ \Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash \textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i = \delta_i}\}\ \textbf{in}\ t_1 \,::\, \tau\,;\sigma}$$

$(\underline{2})$	$\Gamma \,\|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 :: \kappa_2$	(assume)
(7)	$\Gamma,\ r : \%,\ \overline{w_i : \kappa_i} \,\|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 :: \kappa_2$	(Weak. Type Env 2)
(8)	$(\Gamma,\ r : \%,\ \overline{w_i : \kappa_i})[\varphi_2/a] \,\|\, \Sigma$ $\vdash t[\varphi_2/a] :: \tau[\varphi_2/a]\ ;\ \sigma[\varphi_2/a]$	(IH 4 7)
(9)	$\Gamma[\varphi_2/a],\ r : \%,\ \overline{w_i : \kappa_i[\varphi_2/a]} \,\|\, \Sigma$ $\vdash t[\varphi_2/a] :: \tau[\varphi_2/a]\ ;\ \sigma[\varphi_2/a]$	(Def. Sub. 8)
(10)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash_{\mathrm{T}} \delta_i[\varphi_2/a] :: \kappa_i[\varphi_2/a]$	(Sub. Type/Type 6 2)
(11)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash_{\mathrm{K}} \kappa_i[\varphi_2/a] :: \Diamond$	(Sub. Type/Kind 5 2)
(12)	$\overline{\delta_i[\varphi_2/a]}$ well formed	(Def. Well Formed 6)
(13)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash \textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i = \delta_i[\varphi_2/a]}\}\ \textbf{in}\ t[\varphi_2/a]$ $:: \tau[\varphi_2/a]\ ;\sigma[\varphi_2/a]$	(TyLetRegion 9..12)
(14)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash (\textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i = \delta_i}\}\ \textbf{in}\ t)[\varphi_2/a]$ $:: \tau[\varphi_2/a]\ ;\sigma[\varphi_2/a]$	(Def. Sub. 13)

Case: $t = (\textbf{case}\ t\ \textbf{of}\ \overline{K\ x : \tau \to t'})$ / TyCase
Case: $t = K\ \overline{\varphi}\ \overline{t}$ / ...
Case: $t = \mathit{update}_{K,i}\ \overline{\varphi}\ t_1\ t_2$ / TyUpdate
Case: $t = \mathit{suspend}\ \overline{\varphi}\ t_1\ t_2$ / TySuspend
Similarly to TyAppT case.

Case: $t = \textbf{mask}\ \delta\ \textbf{in}\ t$ / TyMask

$$\frac{(3)\ \Gamma,\ a_2 : \kappa_2 \,|\, \Sigma \vdash t \,::\, \tau\,;\sigma \qquad (4)\ \Gamma,\ a_2 : \kappa_2 \,|\, \Sigma \vdash_{\mathrm{T}} \delta \,::\, \mathit{Pure}\ \sigma'}{(\underline{1})\ \Gamma,\ a_2 : \kappa_2 \,|\, \Sigma \vdash \textbf{mask}\ \delta\ \textbf{in}\ t \,::\, \tau\,;\sigma \setminus \sigma'}$$

$(\underline{2})$	$\Gamma \,\|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 :: \kappa_2$	(assume)
(5)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash t[\varphi_2/a] :: \tau[\varphi_2/a]\,;\sigma[\varphi_2/a]$	(IH 3 2)
(6)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash_{\mathrm{T}} \delta[\varphi_2/a] :: (\mathit{Pure}\ \sigma')[\varphi_2/a]$	(Sub. Type/Type 4 2)
(7)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash_{\mathrm{T}} \delta[\varphi_2/a] :: \mathit{Pure}\ \sigma'[\varphi_2/a]$	(Def. Sub. 6)
(8)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash \textbf{mask}\ \delta[\varphi_2/a]\ \textbf{in}\ t[\varphi_2/a]$ $:: \tau[\varphi_2/a]\ ;\ \sigma[\varphi_2/a]/\sigma'[\varphi_2/a]$	(TyMask 5 7)
(9)	$\Gamma[\varphi_2/a] \,\|\, \Sigma \vdash (\textbf{mask}\ \delta\ \textbf{in}\ t)[\varphi_2/a]$ $:: \tau[\varphi_2/a]\ ;\ (\sigma \setminus \sigma')[\varphi_2/a]$	(Def. Sub. 8)

Lemma: (Substitution of Types in Types)

$$\begin{array}{rl} \text{If} & \Gamma,\, a : \kappa_2 \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1 \;::\; \kappa_1 \\ \text{and} & \Gamma \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 \;::\; \kappa_2 \\ \text{then} & \Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1[\varphi_2/a] \;::\; \kappa_1[\varphi_2/a] \end{array}$$

Proof: by induction over the derivation of $\Gamma,\, a : \kappa_2 \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1 \;::\; \kappa_1$

(IH)	Substitution holds for all subterms of φ_1.	(assume)

Case: $\varphi = a$ / KiVar
Similarly to Sub. Value/Value TyVar case.

Case: $\varphi = \forall(b : \kappa_1).\, \tau$ / KiAll

$$\frac{(3)\ \Gamma,\, a : \kappa_2 \,|\, \Sigma \vdash_{\mathrm{K}} \kappa_{11} \;::\; \omega \qquad (4)\ \Gamma,\, a : \kappa_2,\, b : \kappa_{11} \,|\, \Sigma \vdash_{\mathrm{T}} \tau_{12} \;::\; \kappa_1}{(\underline{1})\ \Gamma,\, a : \kappa_2 \,|\, \Sigma \vdash_{\mathrm{T}} \forall(b : \kappa_{11}).\, \tau_{12} \;::\; \kappa_1}$$

($\underline{2}$)	$\Gamma \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 \;::\; \kappa_2$	(assume)
(5)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\mathrm{K}} \kappa_{11}[\varphi_2/a] \;::\; \omega[\varphi_2/a]$	(Sub. Type/Kind 3 2)
(6)	$\Gamma,\, b : \kappa_{11} \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 \;::\; \kappa_2$	(Weak. Type Env 2)
(7)	$(\Gamma,\, b : \kappa_{11})[\varphi_2/a] \,|\, \Sigma \vdash_{\mathrm{T}} \tau_{12}[\varphi_2/a] \;::\; \kappa_1[\varphi_2/a]$	(IH 4 6)
(8)	$\Gamma[\varphi_2/a],\, b : \kappa_{11}[\varphi_2/a] \,|\, \Sigma \vdash_{\mathrm{T}} \tau_{12}[\varphi_2/a] \;::\; \kappa_1[\varphi_2/a]$	(Def. Sub. 7)
(9)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\mathrm{T}} (\forall(b : \kappa_{11}[\varphi_2/a]).\, \tau_{12}[\varphi_2/a]) \;::\; \kappa_1[\varphi_2/a]$	(KiAll 5 8)
(10)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\mathrm{T}} (\forall(b : \kappa_{11}).\, \tau_{12})[\varphi_2/a] \;::\; \kappa_1[\varphi_2/a]$	(Def. Sub. 9)

Case: $\varphi = \varphi_1\ \varphi_2$ / KiApp

$$\frac{(3)\ \Gamma,\, a : \kappa_3 \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1 \;::\; \Pi(b : \kappa_{11}).\, \kappa_{12} \qquad (4)\ \Gamma,\, a : \kappa_3 \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_2 \;::\; \kappa_{11}}{(\underline{1})\ \Gamma,\, a : \kappa_3 \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1\ \varphi_2 \;::\; \kappa_{12}[\varphi_2/b]}$$

($\underline{2}$)	$\Gamma \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_3 \;::\; \kappa_3$	(assume)
(5)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1[\varphi_3/a] \;::\; (\Pi(b : \kappa_{11}).\, \kappa_{12})[\varphi_3/a]$	(IH 4 2)
(6)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1[\varphi_3/a] \;::\; \Pi(b : \kappa_{11}[\varphi_3/a]).\, (\kappa_{12}[\varphi_3/a])$	(Def. Sub. 5)
(7)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_2[\varphi_3/a] \;::\; \kappa_{11}[\varphi_3/a]$	(IH 5 2)
(8)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash_{\mathrm{T}} \varphi_1[\varphi_3/a]\ \varphi_2[\varphi_3/a]$ $::\ (\kappa_{12}[\varphi_3/a])[\varphi_2[\varphi_3/a]/b]$	(KiApp 6 7)
(9)	$\Gamma[\varphi_3/a] \,|\, \Sigma \vdash_{\mathrm{T}} (\varphi_1\ \varphi_2)[\varphi_3/a] \;::\; (\kappa_{12}[\varphi_2/b])[\varphi_3/a]$	(Def. Sub. 8)

The remaining cases are similar to the KiApp case.

Lemma: (Substitution of Types in Kinds)

$$\begin{array}{ll} \text{If} & \Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash_{\text{K}} \kappa_1 \ :: \ \omega \\ \text{and} & \Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \ :: \ \kappa_2 \\ \text{then} & \Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} \kappa_1[\varphi_2/a] \ :: \ \omega[\varphi_2/a] \end{array}$$

Proof: by induction over the derivation of $\Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash_{\text{K}} \kappa_1 \ :: \ \omega$

(IH) Sub. of Types in Kinds holds for all subterms of κ_1. (IH)

Case: $\kappa_1 \in \{*,\ \%,\ !,\ \mathit{Const},\ \mathit{Mutable},\ \mathit{Pure}\}$
Trivial, κ_1 contains no variables.

Case: $\kappa_1 = \kappa_{11}\ \varphi_{12}$ / KsApp

$$\frac{(3)\ \Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash_{\text{K}} \kappa_{11} \ :: \ \kappa_{12} \to \omega \quad (4)\ \Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash_{\text{T}} \varphi_{12} \ :: \ \kappa_{12}}{(\underline{1})\ \Gamma,\ a : \kappa_2 \,|\, \Sigma \vdash_{\text{K}} \kappa_{11}\ \varphi_{12} \ :: \ \omega}$$

($\underline{2}$)	$\Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \ :: \ \kappa_2$	(assume)
(5)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{T}} \kappa_{11}[\varphi_2/a] \ :: \ (\kappa_{12} \to \omega)[\varphi_2/a]$	(IH 3 2)
(6)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{T}} \kappa_{11}[\varphi_2/a] \ :: \ \kappa_{12}[\varphi_2/a] \to \omega[\varphi_2/a]$	(Def. Sub. 5)
(7)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{T}} \varphi_{12}[\varphi_2/a] \ :: \ \kappa_{12}[\varphi_2/a]$	(Sub. Type/Type 4 2)
(8)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} \kappa_{11}[\varphi_2/a]\ \ \varphi_{12}[\varphi_2/a] \ :: \ \omega[\varphi_2/a]$	(KsApp 6 7)
(9)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} (\kappa_{11}\ \varphi_{12})[\varphi_2/a] \ :: \ \omega[\varphi_2/a]$	(Def. Sub 8)

Case: $\kappa_1 = \Pi(b : \kappa_{11}).\ \kappa_{12}$ / KsAbs

$$\frac{(3)\ \Gamma,\ a : \kappa_3 \,|\, \Sigma \vdash_{\text{K}} \kappa_{11} \ :: \ \omega_{11} \quad (4)\ \Gamma,\ a : \kappa_3,\ b : \kappa_{11} \,|\, \Sigma \vdash_{\text{K}} \kappa_{12} \ :: \ \omega_{12}}{(\underline{1})\ \Gamma,\ a : \kappa_3 \,|\, \Sigma \vdash_{\text{K}} \Pi(b : \kappa_{11}).\ \kappa_{12} \ :: \ \omega_{12}}$$

($\underline{2}$)	$\Gamma \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \ :: \ \kappa_2$	(assume)
(5)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} \kappa_{11}[\varphi_2/a] \ :: \ \omega_{11}[\varphi_2/a]$	(IH 3 2)
(6)	$(\Gamma,\ b : \kappa_{11})[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} \kappa_{12}[\varphi_2/a] \ :: \ \omega_{12}[\varphi_2/a]$	(IH 4 2)
(7)	$\Gamma[\varphi_2/a],\ b : \kappa_{11}[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} \kappa_{12}[\varphi_2/a] \ :: \ \omega_{12}[\varphi_2/a]$	(Def. Sub. 6)
(8)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} \Pi(b : \kappa_{11}[\varphi_2/a]).\ \kappa_{12}[\varphi_2/a] \ :: \ \omega_{12}[\varphi_2/a]$	(KsAbs 5 7)
(9)	$\Gamma[\varphi_2/a] \,|\, \Sigma \vdash_{\text{K}} (\Pi(b : \kappa_{11}).\ \kappa_{12})[\varphi_2/a] \ :: \ \omega_{12}[\varphi_2/a]$	(Def. Sub 8)

Theorem: (Progress)

Suppose we have a state $H;t$ with store H and term t. Let Σ be a store typing which models H. If H is well typed, and t is closed and well typed, and t contains no fabricated region witnesses, then either t is a value or $H;t$ can transition to the next state.

$$\begin{array}{ll}
\text{If} & \emptyset \,|\, \Sigma \vdash t \,::\, \tau \,;\, \sigma \\
\text{and} & \Sigma \models H \\
\text{and} & \Sigma \vdash H \\
\text{and} & \text{nofab}(t) \\
\text{then} & t \in \mathit{Value} \\
\text{or} & \text{for some } H', t' \text{ we have} \\
 & \quad (\quad H \,;\, t \longrightarrow H' \,;\, t' \;\text{ and }\; \text{nofab}(t') \\
 & \quad\;\; \text{or } H \,;\, t \longrightarrow H' \,;\, \text{fail})
\end{array}$$

Proof: By induction over the derivation of $\emptyset \,|\, \Sigma \vdash t \,::\, \tau \,;\, \sigma$

Let $(H \,;\, t \text{ can step}) \equiv (\text{for some } H, t \text{ we have } H \,;\, t \longrightarrow H' \,;\, t' \text{ and } \text{nofab}(t'))$

We will not formally prove nofab(t') in the conclusion of each case. This property can be verified by inspecting (EvLetRegion) and noting that unevaluated applications of witness constructors are not substitued into the body of the term.

(IH)	Progress holds for all subterms of t.	(assume)

Case: $t = x$
Can't happen. Type environment is empty.

Case: t is one of l, $()$, $\Lambda(a :: \kappa).\, t'$, $\lambda(x :: \tau).\, t'$
$t \in \mathit{Value}$

Case: $t = (t_1\ \varphi_2)$ / TyAppT

$$\frac{(5)\ \emptyset \,|\, \Sigma \vdash t_1 \,::\, \forall(a : \kappa_{11}).\, \varphi_{12} \,;\, \sigma \quad (6)\ \emptyset \,|\, \Sigma \vdash_{\text{T}} \varphi_2 \,::\, \kappa_{11}}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash t_1\ \varphi_2 \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma[\varphi_2/a]}$$

($\underline{2..4}$)	$\Sigma \models H$, $\Sigma \vdash H$, nofab(t)	(assume)
(7)	$t_1 \in \mathit{Value}$ or $H \,;\, t_1$ can step	(IH 5 2..4)
(8)	*Case:* $t_1 \in \mathit{Value}$	
(9)	$t_1 = \Lambda(a : \kappa_{11}).\, t_{12}$	(Forms of Terms 8 5)
(10)	$H \,;\, t$ can step	(EvTAppAbs 9)
(11)	*Case:* $H \,;\, t_1$ can step	
(12)	$H \,;\, t$ can step	(EvContext 11)

Case: $t = t_1\ t_2$ / TyApp

$$\frac{(5)\ \emptyset\,|\,\Sigma \vdash t_1\ ::\ \tau_{11} \xrightarrow{\sigma} \tau_{12}\,;\sigma_1 \qquad (6)\ \emptyset\,|\,\Sigma \vdash t_2\ ::\ \tau_{11}\,;\sigma_2}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash t_1\ t_2\ ::\ \tau_{12}\,;\sigma_1 \vee \sigma_2 \vee \sigma}$$

$(\underline{2..4})$	$\Sigma \models H,\ \Sigma \vdash H,\ \text{nofab}(t)$	(assume)
(7)	$t_1 \in \mathit{Value}$ or $H\,;t_1$ can step	(IH 5 2..4)
(8)	$t_2 \in \mathit{Value}$ or $H\,;t_2$ can step	(IH 6 2..4)
(9)	*Case* : $H\,;t_1$ can step	
(10)	$H\,;t$ can step	(EvContext 9)
(11, 12)	*Case* : $t_1 \in \mathit{Value},\ H\,;t_2$ can step	
(13)	$H\,;t$ can step	(EvContext 11 12)
(14, 15)	*Case* : $t_1 \in \mathit{Value},\ t_2 \in \mathit{Value}$	
(16)	$t_1 = \lambda(x : \tau_{11}).\,t_{12}$	(Forms of Terms 14 5)
(17)	$H\,;t$ can step	(EvAppAbs 16 15)

Case: $t =$ (**letregion** r **with** $\{\overline{w_i = \delta_i}\}$ **in** t_1) / TyLetRegion

$(\underline{1})$	$\emptyset\,|\,\Sigma \vdash$ (**letregion** r **with** $\{\overline{w_i = \delta_i}\}$ **in** t_1) $::\ \tau_1\,;\sigma_1$	(assume)
$(\underline{2..4})$	$\Sigma \models H,\ \Sigma \vdash H,\ \text{nofab}(t)$	(assume)
(5)	$\overline{\delta_i}$ well formed	(Inv. 1)
(6)	$\delta \in \{\mathit{MkMutable}\ r,\ \mathit{MkConst}\ r\}$	(Def. Well Formed 5)
(7)	$H,\ \overline{\text{propOf}(\Delta_i)}\ ;\ \overline{\delta_i[\underline{\rho}/r] \rightsquigarrow \Delta_i}$	(Region Witness Assertion 6)
(8)	$H\,;t$ can step	(EvLetRegion 7)

Case: $t = K\ \varphi\ \overline{\varphi'}\ \overline{t'}$ / TyData

$$\frac{\begin{array}{c}\overline{(7)\ \emptyset\,|\,\Sigma \vdash t'_i\ ::\ \tau_i[\varphi/r][\overline{\varphi'/a}]\,;\sigma_i}^{\,i\leftarrow 0..n} \\ (5)\ \emptyset\,|\,\Sigma \vdash_{\mathrm{T}} \varphi\ ::\ \%\quad (6)\ K :: \forall(r:\%).\forall(\overline{a:\kappa}).\overline{\tau} \to T\ r\ \overline{a} \in \text{ctorTypes}(T)\end{array}}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash K\ \varphi\ \overline{\varphi'}\ \overline{t'}\ ::\ T\ \varphi\ \overline{\varphi'}\,;\sigma_0 \vee \sigma_1 \ldots \vee \sigma_n}$$

$(\underline{2..4})$	$\Sigma \models H,\ \Sigma \vdash H,\ \text{nofab}(t)$	(assume)
(8)	*Case* : $H\,;t'_i$ can step for some i	
(9)	$H\,;t$ can step	(EvContext 8)
(10)	*Case* : forall i we have $t'_i \in \mathit{Value}$	
(11)	$\varphi = \underline{\rho}$, for some ρ	(Forms of Types 5)
(12)	$\underline{\rho} \in \Sigma$	(KiHandle 5 11)
(13)	$\rho \in H$	(Def. Store Model 2 12)
(14)	$H\,;t$ can step	(EvAlloc 10 11 13)

Case: $t = (\textbf{case}\ t_1\ \textbf{of}\ \overline{p \to t'})$ / TyCase

$$\frac{(5)\ \emptyset \,|\, \Sigma \vdash t_1 :: T\ \varphi\ \overline{\varphi'}\ ;\ \sigma \qquad (6)\ \overline{\emptyset \,|\, \Sigma \vdash p_i \to t'_i :: T\ \varphi\ \overline{\varphi'} \to \tau\ ;\ \sigma'_i}^{\,i \leftarrow 0..n}}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash \textbf{case}\ t_1\ \textbf{of}\ \overline{p \to t'} :: \tau\ ;\ \sigma \vee Read\ \varphi \vee \sigma'_0 \vee \sigma'_1 \ldots \vee \sigma'_n}$$

($\underline{2..4}$)	$\Sigma \models H,\ \Sigma \vdash H,\ \text{nofab}(t)$	(assume)
(7)	$t_1 \in Value$ or $H\,;t_1$ can step	(IH 5 2..4)
(8)	$Case:\ H\,;t_1$ can step	
(9)	$H\,;t$ can step	(EvContext 8)
(10)	$Case:\ t_1 \in Value$	
(11)	$t_1 = l$	(Forms of Terms 10 5)
(12)	$\emptyset \,|\, \Sigma',\ l : T\ \rho\ \overline{\varphi'} \vdash l :: T\ \rho\ \overline{\varphi'}\ ;\ \bot$	(5, 11)
(13)	$l \overset{\rho}{\mapsto} C_K\ \overline{v^\circ} \in H$	(Def. Store Model 2 12)
(14)	$H\,;t$ can step	(EvCase 13)

Case: $t_1 = (update_{K,i}\ \varphi\ \overline{\varphi'}\ \delta\ t_2\ t_3)$ / TyUpdate

$$\frac{\begin{array}{ll} (6)\ \emptyset \,|\, \Sigma \vdash_{\scriptscriptstyle T} \delta :: Mutable\ \varphi & (8)\ \emptyset \,|\, \Sigma \vdash t_3 :: \tau_i[\varphi/r][\overline{\varphi'/a}]\ ;\ \sigma' \\ (5)\ \emptyset \,|\, \Sigma \vdash t_2 :: T\ \varphi\ \overline{\varphi'}\ ;\ \delta & (7)\ K :: \forall(r:\%).\forall(\overline{a:\kappa}).\overline{\tau} \to T\ r\ \overline{a} \in \text{ctorTypes}(T) \end{array}}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash (update_{K,i}\ \varphi\ \overline{\varphi'}\ \delta\ t_2\ t_3) :: ()\ ;\ (\sigma \vee \sigma' \vee Write\ \varphi)}$$

($\underline{2..4}$)	$\Sigma \models H,\ \Sigma \vdash H,\ \text{nofab}(t)$	(assume)
(9)	$\delta = \underline{\text{mutable}\ \rho}$ for some ρ	(Forms of Types 6 4)
(10, 11)	$\emptyset \,|\, \Sigma \vdash_{\scriptscriptstyle T} \underline{\text{mutable}\ \rho} :: Mutable\ \underline{\rho},\quad \varphi = \underline{\rho}$	(KiMutable 9 6)
(12)	$\underline{\text{mutable}\ \rho} \in \Sigma$	(KiMutable 10)
(13)	$Case:\ H\,;t_2$ can step	
(14)	$H\,;t_1$ can step	(EvContext 13)
(15, 16)	$Case:\ t_2 \in Value,\ H\,;t_3$ can step	
(17)	$t_2 = l$	(Forms of Terms 15 5)
(18)	$H\,;t_1$ can step	(EvContext 17 16)
(19, 20)	$Case:\ t_1 \in Value,\ t_2 \in Value$	
(21)	$t_2 = l$	(Forms of Terms 19 5)
(22)	$\emptyset \,|\, \Sigma \vdash l :: T\ \underline{\rho}\ \overline{\varphi'}\ ;\ \bot$	(TyLoc 5 11)
(23)	$l :: T\ \underline{\rho}\ \overline{\varphi'} \in \Sigma$	(TyLoc 22)
(24)	$l \overset{\rho}{\mapsto} C_{K'}\ \overline{v^\circ} \in H$ for some $K', \overline{v^\circ}$	(Def. Store Model 2 23)
(25)	$\underline{\text{mutable}\ \rho} \in H$	(Def. Store Model 2 12)
(26)	$Case:\ K = K'$	
(27)	$H\,;t$ can step	(EvUpdateBind 25 24 9 21 20)
(28)	$Case:\ K \neq K'$	
(29)	$H;t \longrightarrow H';\text{fail}$	(EvUpdateFail 25 24 9 21 20)

Case: $t = (\mathit{suspend}\ \delta\ t_1\ t_2)$ / TySuspend

$$\frac{(5)\ \emptyset\,|\,\Sigma \vdash_{\mathrm{T}} \delta :: \mathit{Pure}\ \sigma \quad (6)\ \emptyset\,|\,\Sigma \vdash t_1 :: \tau_{11} \xrightarrow{\sigma} \tau_{12}\,;\,\sigma_1 \quad (7)\ \emptyset\,|\,\Sigma \vdash t_2 :: \tau_{11}\,;\,\sigma_2}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash (\mathit{suspend}\ \tau_{11}\ \tau_{12}\ \sigma\ \delta\ t_1\ t_2) :: \tau_{12}\,;\,\sigma_1 \vee \sigma_2}$$

($\underline{2..4}$)	$\Sigma \models H,\ \Sigma \vdash H,\ \mathrm{nofab}(t)$	(assume)
(8)	$\delta \in \{\mathit{MkPurify}\ \delta_2,\ \mathit{MkPureJoin}\ \delta_3\ \delta_4,\ \mathit{MkPureBot},$ $\underline{\mathrm{pure}\ \sigma}\}$	(Forms of Types 5)
(9)	$\mathit{Case}:\ \delta \in \{\mathit{MkPurify}\ \delta_2,\ \mathit{MkPureJoin}\ \delta_3\ \delta_4,\ \mathit{MkPureBot}\}$	
(10)	$H\ ;\ \delta \rightsquigarrow \delta'$	(Progress of Purity 5 9 4)
(11)	$H\,;\,t$ can step	(EvSuspendWit 10)
(12..14)	$\mathit{Case}:\ \delta = \underline{\mathrm{pure}\ \sigma},\ t_1 \in \mathit{Value},\ t_2 \in \mathit{Value}$	
(15)	$t_1 = \lambda(x:\tau).\,t_3$	(Forms of Terms 6 13)
(16)	$H\,;\,t$ can step	(EvSuspendApp 12 15 14)

Case: $t = \textbf{mask}\ \delta\ \textbf{in}\ t_1$

$$\frac{(5)\ \emptyset\,|\,\Sigma \vdash t :: \tau\,;\,\sigma \quad (6)\ \emptyset\,|\,\Sigma \vdash_{\mathrm{T}} \delta :: \mathit{Pure}\ \sigma'}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash \textbf{mask}\ \delta\ \textbf{in}\ t :: \tau\,;\,\sigma \setminus \sigma'}$$

($\underline{2..4}$)	$\Sigma \models H,\ \Sigma \vdash H,\ \mathrm{nofab}(t)$	(assume)
(7)	$\delta \in \{\mathit{MkPurify}\ \delta_2,\ \mathit{MkPureJoin}\ \delta_3\ \delta_4,\ \mathit{MkPureBot},$ $\underline{\mathrm{pure}\ \sigma}\}$	(Forms of Types 6)
(8)	$\mathit{Case}:\ \delta \in \{\mathit{MkPurify}\ \delta_2,\ \mathit{MkPureJoin}\ \delta_3\ \delta_4,\ \mathit{MkPureBot}\}$	
(9)	$H\ ;\ \delta \rightsquigarrow \delta'$	(Progress of Purity 5 8 4)
(10)	$H\,;\,t$ can step	(EvMaskWit 10)
(11)	$\mathit{Case}:\ \delta = \underline{\mathrm{pure}\ \sigma}$	
(12)	$H\,;\,t$ can step	(EvMaskApp 11)

The remaining cases are similar to the TyApp case.

Theorem: (Preservation)

Suppose we have a state $H;t$ with store H and closed term t. Let Σ be a store typing which models H. If H and t are well typed, and $H;t$ can transition to a new state $H';t'$ then for some Σ' which models H', H' is well typed, t' has the same type as t, and t' has an effect that is no greater than for t.

$$\begin{array}{ll}
\text{If} & \emptyset \,|\, \Sigma \vdash t \,::\, \tau \,;\, \sigma \\
\text{and} & H \,;\, t \longrightarrow H' \,;\, t' \\
\text{and} & \Sigma \models H \ \text{ and } \ \Sigma \vdash H \\
\multicolumn{2}{l}{\text{then for some } \Sigma',\ \sigma' \text{ we have}} \\
 & \emptyset \,|\, \Sigma' \vdash t' \,::\, \tau \,;\, \sigma' \\
\text{and} & \Sigma' \supseteq \Sigma \ \text{ and } \ \Sigma' \models H' \ \text{ and } \ \Sigma' \vdash H' \\
\text{and} & \sigma' \sqsubseteq_{\Sigma'} \sigma
\end{array}$$

Proof: By induction over the derivation of $\emptyset \,|\, \Sigma \vdash t \,::\, \tau \,;\, \sigma$.

(IH) Preservation holds for all subterms of t. (assume)

Case: $t = x$
Can't happen. Type environment is empty.

Case: t is one of l, $()$, $\Lambda(a :: \kappa).\, t'$, $\lambda(x :: \tau).\, t'$
Can't happen. There is no transition rule for $H;t$

Case: $t = t_1\ \varphi_2$ / TyAppT / EvContext

$$\frac{(5)\ \emptyset \,|\, \Sigma \vdash t_1 \,::\, \forall(a : \kappa_{11}).\, \varphi_{12} \,;\, \sigma \qquad (6)\ \emptyset \,|\, \Sigma \vdash_{\scriptscriptstyle T} \varphi_2 \,::\, \kappa_{11}}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash t_1\ \varphi_2 \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma[\varphi_2/a]}$$

$$\frac{(7)\ H \,;\, t_1 \longrightarrow H' \,;\, t_1'}{(\underline{2})\ H \,;\, t_1\ \varphi_2 \longrightarrow H' \,;\, t_1'\ \varphi_2}$$

($\underline{3}..\underline{4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(8..12)	$\emptyset \,|\, \Sigma' \vdash t_1' \,::\, \forall(a : \kappa_{11}).\, \varphi_{12} \,;\, \sigma'$, $\Sigma' \supseteq \Sigma,\ \Sigma' \models H',\ \Sigma' \vdash H',\ \sigma' \sqsubseteq_{\Sigma'} \sigma$	(IH 5 7 3 4)
(13)	$\emptyset \,|\, \Sigma' \vdash_{\scriptscriptstyle T} \varphi_2 \,::\, \kappa_{11}$	(Weak. Store Typing 6 9)
(14)	$\emptyset \,|\, \Sigma' \vdash t_1'\ \varphi_2 \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma'[\varphi_2/a]$	(TyAppT 8 13)
(15)	$\sigma'[\varphi_2/a] \sqsubseteq_{\Sigma'} \sigma[\varphi_2/a]$	(Def. Sub, $\sqsubseteq$ 12)

Case: $t = t_1\ \varphi_2$ / TyAppT / EvTAppAbs

$$\frac{\dfrac{(7)\ a : \kappa_{11} \,|\, \Sigma \vdash t_{12} \,::\, \varphi_{12} \,;\, \sigma}{(5)\ \emptyset \,|\, \Sigma \vdash \Lambda(a : \kappa_{11}).t_{12} \,::\, \forall(a : \kappa_{11}).\, \varphi_{12} \,;\, \sigma} \qquad (6)\ \emptyset \,|\, \Sigma \vdash_{\scriptscriptstyle T} \varphi_2 \,::\, \kappa_{11}}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash (\Lambda(a : \kappa_{11}).t_{12})\ \varphi_2 \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma[\varphi_2/a]}$$

$$(\underline{2})\ H; (\Lambda(a :: \kappa_{11}).\, t_{12})\ \varphi_2 \longrightarrow H; t_{12}[\varphi_2/a]$$

($\underline{3}..\underline{4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(9)	$\emptyset \,|\, \Sigma \vdash t_{12}[\varphi_2/a] \,::\, \varphi_{12}[\varphi_2/a] \,;\, \sigma[\varphi_2/a]$	(Sub. Type/Value 7 6)

Case: $t = t_1\ t_2$ / TyApp / EvContext ($E_v\ t_2$)

$$\frac{(5)\ \emptyset\,|\,\Sigma \vdash t_1 :: \tau_{11} \xrightarrow{\sigma} \tau_{12}\,;\, \sigma_1 \qquad (6)\ \emptyset\,|\,\Sigma \vdash t_2 :: \tau_{11}\,;\,\sigma_2}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash t_1\ t_2 :: \tau_{12}\,;\,\sigma_1 \vee \sigma_2 \vee \sigma}$$

$$\frac{(7)\ H\,;\,t_1 \longrightarrow H'\,;\,t_1'}{(\underline{2})\ H\,;\,t_1\ t_2 \longrightarrow H'\,;\,t_1'\ t_2}$$

($\underline{3..4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(8..12)	$\emptyset\,|\,\Sigma' \vdash t_1' :: \tau_{11} \xrightarrow{\sigma} \tau_{12}\,;\,\sigma_1'$	
	$\Sigma' \supseteq \Sigma,\ \Sigma' \models H',\ \Sigma' \vdash H',\ \sigma_1' \sqsubseteq_{\Sigma'} \sigma_1$	(IH 5 7 3 4)
(13)	$\emptyset\,|\,\Sigma' \vdash t_2 :: \tau_{11}\,;\,\sigma_2$	(Weak. Store Typing 6 9)
(14)	$\emptyset\,|\,\Sigma' \vdash t_1'\ t_2 :: \tau_{12}\,;\,\sigma_1' \vee \sigma_2 \vee \sigma$	(TyApp 8 13)
(15)	$\sigma_1' \vee \sigma_2 \vee \sigma\ \sqsubseteq_{\Sigma'}\ \sigma_1 \vee \sigma_2 \vee \sigma$	(Def. $\sqsubseteq$ 12)

Case: $t = t_1\ t_2$ / TyApp / EvContext ($v\ E_v$)
Similarly to TyApp / EvContext ($E_v\ t_2$) case.

Case: $t = t_1\ t_2$ / TyApp / EvAppAbs

$$\frac{\dfrac{(7)\ x : \tau_{11}\,|\,\Sigma \vdash t_{12} :: \tau_{12}\,;\,\sigma}{(5)\ \emptyset\,|\,\Sigma \vdash \lambda(x : \tau_{11}).\,t_{12} :: \tau_{11} \xrightarrow{\sigma} \tau_{12}\,;\,\sigma_1} \qquad (6)\ \emptyset\,|\,\Sigma \vdash v^\circ :: \tau_{11}\,;\,\bot}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash (\lambda(x : \tau_{11}).\,t_{12})\ v^\circ :: \tau_{12}\,;\,\sigma_1 \vee \sigma}$$

$$(\underline{2})\ H\,;\,(\lambda(x : \tau_{11}).\,t_{12})\ v^\circ \longrightarrow H'\,;\,t_{12}[v^\circ/x]$$

($\underline{3..4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(8)	$\emptyset\,|\,\Sigma \vdash t_{12}[v^\circ/x] :: \tau_{12}\,;\,\sigma$	(Sub. Value/Value 7 6)
(9)	$\sigma \sqsubseteq_{\Sigma'} \sigma_1 \vee \sigma$	(Def. $\sqsubseteq$)

Case: $t = K\ \underline{\rho}\ \overline{\varphi}\ \overline{v^\circ}$ / TyData / EvAlloc

$$\frac{(5)\ \emptyset\,|\,\Sigma \vdash_{\mathrm{T}} \underline{\rho} :: \% \quad (6)\ K :: \forall(r : \%).\forall(\overline{a : \kappa}).\overline{\tau} \to T\ r\ \overline{a} \in \mathrm{ctorTypes}(T) \quad \overline{(7)\ \emptyset\,|\,\Sigma \vdash v_i^\circ :: \tau_i[\underline{\rho}/r][\overline{\varphi/a}]\,;\,\bot}^{\,i \leftarrow 0..n}}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash K\ \underline{\rho}\ \overline{\varphi}\ \overline{v^\circ} :: T\ \underline{\rho}\ \overline{\varphi}\,;\,\bot}$$

$$(\underline{2})\ H[\rho]\,;\,K\ \underline{\rho}\ \overline{\varphi}\ \overline{v^\circ} \longrightarrow H,\ l \overset{\rho}{\mapsto} C_K\ \overline{v^\circ}\,;\,l$$

($\underline{3..4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(8)	let $\Sigma' = \Sigma,\ l :: T\ \underline{\rho}\ \overline{\varphi}$	
(9)	let $H' = H,\ l \overset{\rho}{\mapsto} C_K\ \overline{v^\circ}$	
(10)	$\Gamma\,|\,\Sigma' \vdash l :: T\ \underline{\rho}\ \overline{\varphi}\,;\,\bot$	(TyLoc 8)
(11)	$\Sigma' \supseteq \Sigma$	(Def. $\supseteq$ 8)
(12)	$\Sigma' \models H'$	(Store Model 3 8 9 6 7)
(13)	$\Sigma' \vdash H'$	(Well Typed Store 4 9 8 6 7)
(14)	$\bot \sqsubseteq_{\Sigma'} \bot$	(Def. $\sqsubseteq$)

Case: $t = (\textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i = \delta_i}\}\ \textbf{in}\ t_1)$ / TyLetRegion / EvLetRegion

$$\frac{\begin{array}{lll}(5)\ \overline{\delta_i}\ \text{well formed} & (6)\ \overline{\emptyset \,|\, \Sigma \vdash_{\scriptscriptstyle T} \delta_i :: \kappa_i}^{\,i} & (7)\ \overline{\emptyset \,|\, \Sigma \vdash_{\scriptscriptstyle K} \kappa_i :: \Diamond}^{\,i} \\ (8)\ r \notin fv(\tau) & (9)\ r : \%,\ \overline{w_i : \kappa_i} \,|\, \Sigma \vdash t_1 :: \tau\,;\sigma & \end{array}}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash (\textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i = \delta_i}\}\ \textbf{in}\ t_1) :: \tau\,;\sigma}$$

$$\frac{(10)\ \overline{H,\ \underline{\rho},\ \text{propOf}(\Delta_i)\ ;\ \delta_i[\underline{\rho}/r] \rightsquigarrow \Delta_i}^{\,i} \qquad (11)\ \underline{\rho}\ \text{fresh}}{(\underline{2})\ H\,;\textbf{letregion}\ r\ \textbf{with}\ \{\overline{w_i = \delta_i}\}\ \textbf{in}\ t_1 \longrightarrow H,\ \underline{\rho},\ \overline{\text{propOf}(\Delta_i)}\ ;\ t_1[\underline{\rho}/r\ \overline{\Delta_i/w_i}]}$$

$(\underline{3..4})$	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(12)	let $H' = H,\ \underline{\rho},\ \overline{\text{propOf}(\Delta_i)}$	(let)
(13)	let $\Sigma' = \Sigma,\ \underline{\rho},\ r \sim \underline{\rho},\ \overline{\Delta_i}$	(let)
(14)	$\Sigma' \models H'$	(Store Model 3 13 12)
(15)	$\Sigma' \vdash H'$	(Well Typed Store 4 12 13)
(16)	$r : \%,\ \overline{w_i : \kappa_i} \,|\, \Sigma' \vdash t_1 :: \tau\,;\sigma$	(Weak. Store Typing 9 13)
(17)	$\emptyset \,|\, \Sigma' \vdash_{\scriptscriptstyle T} \underline{\rho} :: \%$	(KiHandle 13)
(18)	$\overline{\emptyset \,|\, \Sigma' \vdash_{\scriptscriptstyle T} \Delta_i :: \kappa_i}^{\,i}$	(KiConst, KiMutable, 13, 5)
(19)	$\emptyset \,|\, \Sigma' \vdash t_1[\underline{\rho}/r\ \overline{\Delta_i/w_i}] :: \tau[\underline{\rho}/r\ \overline{\Delta_i/w_i}]\,;\sigma[\underline{\rho}/r\ \overline{\Delta_i/w_i}]$	(Subst. Type/Value 5 17 18)
(20)	$\tau[\underline{\rho}/r\ \overline{\Delta_i/w_i}] \equiv \tau[\underline{\rho}/r]$	(No Wit. Vars in Value Types)
(21)	$\sigma[\underline{\rho}/r\ \overline{\Delta_i/w_i}] \equiv \sigma[\underline{\rho}/r]$	(No Wit. Vars in Effect Types)
(22)	$\emptyset \,|\, \Sigma' \vdash t_1[\underline{\rho}/r\ \overline{\Delta_i/w_i}] :: \tau[\underline{\rho}/r]\,;\sigma[\underline{\rho}/r]$	(Equiv. 19 20 21)
(23)	$\emptyset \,|\, \Sigma' \vdash t_1[\underline{\rho}/r\ \overline{\Delta_i/w_i}] :: \tau\,;\sigma[\underline{\rho}/r]$	(Def. Sub. 22 8)
(24)	$\sigma[\underline{\rho}/r] \sqsubseteq_{\Sigma'} \sigma$	(Def. $\sqsubseteq$ 13, Region Phase Change)

Case: $t = \textbf{case}\ l\ \textbf{of}\ ...\ K\ \overline{x}\ ...$ / TyCase / EvCase

$$\frac{(5)\ \emptyset \,|\, \Sigma \vdash l :: T\ \underline{\rho}\ \overline{\varphi'}\,;\sigma \quad (6)\ \overline{\emptyset \,|\, \Sigma \vdash K_i\ \overline{x} \to t_i :: T\ \underline{\rho}\ \overline{\varphi'} \to \tau\,;\sigma'_i}^{\,i \leftarrow 0..n}}{(1)\ \emptyset \,|\, \Sigma \vdash \textbf{case}\ l\ \textbf{of}\ ...\ K_j\ \overline{x} \to t_j\ ... :: \tau\,;\sigma \vee \mathit{Read}\ \underline{\rho} \vee \sigma'_0 \vee \sigma'_1 ... \vee \sigma'_n}$$

$$(2)\ H[l \overset{\rho}{\mapsto} C_K\ \overline{v^\circ}]\,;\textbf{case}\ l\ \textbf{of}\ ...\ K_j\ \overline{x} \to t_j\ ... \longrightarrow H\,;t_j[\overline{v^\circ/x}]$$

$(\underline{3..4})$	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(7, 8)	$\emptyset \,|\, \Sigma \vdash K_j\ \overline{x_k}^{\,k} \to t_j :: T\ \underline{\rho}\ \overline{\varphi} \to \tau\,;\sigma'_j,\quad j \in 0..n$	(Exhaustive Cases 6)
(9)	$K :: \forall(r : \%)(\overline{a : \kappa}).\overline{\tau'} \to T\ r\ \overline{a} \in \text{ctorTypes}(T)$	(Inv. TyAlt 7)
(10)	$\overline{x_k : \tau'_k[\underline{\rho}/r\ \overline{\varphi/a}]}^{\,k} \,|\, \Sigma \vdash t_j :: \tau\,;\sigma'_j$	(Inv. TyAlt 7)
(11)	$l :: T\ \underline{\rho}\ \overline{\varphi} \in \Sigma$	(Well Typed Store, TyLoc 4 2 5)
(12)	$\overline{\emptyset \,|\, \Sigma \vdash v^\circ_k :: \tau'_k[\underline{\rho}/r\ \overline{\varphi/a}]\,;\bot}^{\,k}$	(Well Typed Store 4 12)
(13)	$\emptyset \,|\, \Sigma \vdash t_j[\overline{v^\circ_k/x}^{\,k}] :: \tau\,;\sigma'_j$	(Sub. Value/Value 10 12)
(14)	$\sigma'_j \sqsubseteq_{\Sigma'} \sigma \vee \mathit{Read}\ \underline{\rho} \vee \sigma'_0 \vee \sigma'_1 ... \vee \sigma'_n$	(Def. $\sqsubseteq$ 8)

Case: $t = update_{K,i}\ \underline{\rho}\ \overline{\varphi}\ \underline{mutable\ \rho}\ l\ u^{\circ}$ / TyUpdate / EvUpdateBind

$$\frac{\begin{array}{ll}(5)\ \emptyset\,|\,\Sigma \vdash_{\mathrm{T}} \underline{\mathrm{mutable}\ \rho}\ ::\ Mutable\ \underline{\rho} & (6)\ \emptyset\,|\,\Sigma \vdash u^{\circ}\ ::\ \tau_i[\underline{\rho}/r\ \overline{\varphi/a}]\ ;\ \bot \\ (7)\ \emptyset\,|\,\Sigma \vdash l\ ::\ T\ \underline{\rho}\ \overline{\varphi}\ ;\ \bot & (8)\ K :: \forall(r:\%)(\overline{a:\kappa}).\overline{\tau} \to T\ r\ \overline{a} \in \mathrm{ctorTypes}(T)\end{array}}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash update_{K,i}\ \underline{\rho}\ \overline{\varphi}\ \underline{\mathrm{mutable}\ \rho}\ l\ u^{\circ}\ ::\ ()\ ;\ Write\ \underline{\rho}}$$

$$(\underline{2})\quad \begin{array}{lll} & H[\mathrm{mutable}\ \underline{\rho},\ l \overset{\rho}{\mapsto} C_K\ \overline{v^{\circ}}] & ;\ update_{K,i}\ \underline{\rho}\ \overline{\varphi}\ \underline{\mathrm{mutable}\ \rho}\ l\ u^{\circ} \\ \longrightarrow & H,\ l \overset{\rho}{\mapsto} C_K\ v^{\circ}_0..u^{\circ}_i...v^{\circ}_n & ;\ ()\end{array}$$

($\underline{3..4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(9)	$\emptyset\,|\,\Sigma \vdash ()\ ::\ ()\ ;\ \bot$	(TyUnit)
(10)	$\bot \sqsubseteq_{\Sigma} Write\ \underline{\rho}$	(Def. $\sqsubseteq$)
(11)	let $H' = H,\ l \overset{\rho}{\mapsto} C_K\ v^{\circ}_0..u^{\circ}_i...v^{\circ}_n$	(let)
(12)	$\overline{\emptyset\,|\,\Sigma \vdash v^{\circ}_j\ ::\ \tau_j[\underline{\rho}/r\ \overline{\varphi/a}]\ ;\ \bot}^{\,j\leftarrow 0..n}$	(Well Typed Store 4 11)
(13)	$\Sigma \vdash H'$	(Well Typed Store 12 6 4 11)
(14)	$\Sigma \models H'$	(Store Model 4 11)

Case: $t = suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \delta\ t_1\ t_2$ / TySuspend / EvSuspendWit

$$\frac{\begin{array}{ll} & (5)\ \emptyset\,|\,\Sigma \vdash_{\mathrm{T}} \delta\ ::\ Pure\ \sigma \\ (3)\ \emptyset\,|\,\Sigma \vdash t_1\ ::\ \tau_{11} \xrightarrow{\sigma} \tau_{12}\ ;\ \sigma_1 & (4)\ \emptyset\,|\,\Sigma \vdash t_2\ ::\ \tau_{11}\ ;\ \sigma_2\end{array}}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \delta\ t_1\ t_2\ ::\ \tau_{12}\ ;\ \sigma_1 \vee \sigma_2}$$

$$\frac{(6)\ H\,;\ \delta \rightsquigarrow \delta'}{(2)\ H\,;\ suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \delta\ t_1\ t_2 \longrightarrow H\,;\ suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \delta'\ t_1\ t_2}$$

($\underline{3..4}$)	$\Sigma \models H, \Sigma \vdash H$	(assume)
(7)	$\emptyset\,|\,\Sigma \vdash \delta'\ ::\ Pure\ \sigma\,;$	(Pres. Purity 5 6)
(8)	$\emptyset\,|\,\Sigma \vdash suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \delta'\ t_1\ t_2\ ::\ \tau_{12}\ ;\ \sigma_1 \vee \sigma_2$	(TySuspend 3 7 4)

Case: $t = suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \underline{\mathrm{pure}\ \sigma}\ (\lambda(x:\tau_{11}).\,t_{12})\ v^{\circ}_2$ / TySuspend / EvSuspendApp

$$\frac{\begin{array}{ll}\dfrac{(6)\ x:\tau_{11}\,|\,\Sigma \vdash t_{12}\ ::\ \tau_{12}\ ;\ \sigma}{(5)\ \emptyset\,|\,\Sigma \vdash \lambda(x:\tau_{11}).\,t_{12}\ ::\ \tau_{11} \xrightarrow{\sigma} \tau_{12}\ ;\ \bot} & \begin{array}{l}(7)\ \emptyset\,|\,\Sigma \vdash_{\mathrm{T}} \underline{\mathrm{pure}\ \sigma}\ ::\ Pure\ \sigma \\ (8)\ \emptyset\,|\,\Sigma \vdash v^{\circ}_2\ ::\ \tau_{11}\ ;\ \bot\end{array}\end{array}}{(\underline{1})\ \emptyset\,|\,\Sigma \vdash suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \underline{\mathrm{pure}\ \sigma}\ (\lambda(x:\tau_{11}).\,t_{12})\ v^{\circ}_2\ ::\ \tau_{12}\ ;\ \bot}$$

$$(\underline{2})\ H\,;\ suspend\ \tau_{11}\ \tau_{12}\ \sigma\ \underline{\mathrm{pure}\ \sigma}\ (\lambda(x:\tau_{11}).\,t_{12})\ v^{\circ}_2 \longrightarrow H\,;\ t_{12}[v^{\circ}_2/x]$$

($\underline{3..4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(9)	$\emptyset\,|\,\Sigma \vdash t_{12}[v^{\circ}_2/x]\ ::\ \tau_{12}\ ;\ \sigma$	(Sub. Value/Value 6 8)
(10)	$\sigma \sqsubseteq_{\Sigma} \bot$	(ObsPure 7)

Case: $t = \textbf{mask}\ \delta\ \textbf{in}\ t_1$ / TyMask / EvMaskWit
Similarly to TySuspend / EvSuspendWit case.

Case: $t = \textbf{mask}\ \underline{\text{pure}\ \sigma'}\ \textbf{in}\ t_1$ / TyMask / EvMaskApp

$$\frac{(5)\ \emptyset \,|\, \Sigma \vdash t_1 \ ::\ \tau\,;\sigma \qquad (6)\ \emptyset \,|\, \Sigma \vdash_{\mathrm{T}} \underline{\text{pure}\ \sigma'}\ ::\ \mathit{Pure}\ \sigma'}{(\underline{1})\ \emptyset \,|\, \Sigma \vdash \textbf{mask}\ \underline{\text{pure}\ \sigma'}\ \textbf{in}\ t_1 \ ::\ \tau\,;\sigma \setminus \sigma'}$$

$$(\underline{2})\ H\,;\textbf{mask}\ \underline{\text{pure}\ \sigma'}\ \textbf{in}\ t_1 \longrightarrow H\,;t$$

($\underline{3..4}$)	$\Sigma \models H,\ \Sigma \vdash H$	(assume)
(7)	$\sigma' \sqsubseteq_\Sigma \bot$	(ObsPure 6)
(8)	$\sigma\ \sqsubseteq_\Sigma \sigma \setminus \sigma'$	(Prop. of $\sqsubseteq$ 7)

The remaining cases are similar to the TyApp / EvContext ($E_v\ t_2$) case.