

Sound Mutable and Const Types

SRL Technical Report SRL2008-03

Swaroop Sridhar Jonathan S. Shapiro Scott F. Smith

Department of Computer Science

The Johns Hopkins University

3400 N.Charles Street. 224 NEB. Baltimore, MD 21218.

swaroop@cs.jhu.edu shap@eros-os.org scott@cs.jhu.edu

Abstract

Subsequent to the work presented in Sound and Complete Type Inference in BitC, we introduced by-reference parameters and discovered that the definition of mutability described in that report was insufficient. The language specification was updated to have a path-wise notion of mutability. At the same time, a `const` meta-constructor was introduced into the language to strip shallow mutability (up to the ref boundary) from existing types.

The following type rules reflect the modifications to the language associated with that change.

Language grammar:

Identifiers	$x ::= y \mid z \mid \dots$
Booleans	$b ::= true \mid false$
Indices	$i ::= 1 \mid 2 \mid !i$
Opt. Const	$e ::= e \mid [e]$
Values	$v ::= () \mid b \mid \lambda x.e \mid (v, v) \mid \ell$
Syn. Value	$v ::= v \mid x \mid 1 \mid (v, v)$
Left Expr	$l ::= x \mid 1 \mid e^\wedge \mid l.i$
Expressions	$e ::= v \mid e \mid e \mid l := e \mid \text{if } e \text{ then } e \text{ else } e$ $\mid \text{dup}(e) \mid e^\wedge \mid (e, e) \mid e.i$ $\mid \text{let}^\kappa x = e \text{ in } e$
Let-kinds	$\kappa ::= - \mid \kappa \mid \psi \mid \forall$
Locations	$L ::= 1 \mid \ell$

Types grammar:

Type Variables	$\alpha ::= \alpha \mid \beta \mid \gamma \mid \delta \mid \varepsilon \mid \dots$
M-Vars	$\varsigma ::= \alpha \mid \Psi\alpha$
Types.1	$\rho ::= \alpha \mid \text{unit} \mid \text{bool} \mid \tau \rightarrow \tau \mid \tau \times \tau$
Ref/Pointer	$\mid \uparrow\tau$
Mutable, const	$\mid \Psi\rho \mid [\tau]$
Types.2	$\varrho ::= \rho \mid \alpha \downarrow \rho$
Types	$\tau ::= \varrho \mid \varsigma \downarrow \rho$
Type Scheme	$\sigma ::= \tau \mid \forall \bar{\alpha}. \tau \setminus \mathcal{D}$
Poly. Constraints	$d ::= \star_x^\kappa(\tau)$
Poly. Constraint Sets	$\mathcal{D} ::= \emptyset \mid \{\bar{d}\} \mid \mathcal{D} \cup \mathcal{D}$

Grammar for Dynamic Semantics:

Stack	$S ::= \emptyset \mid S, l \mapsto v$
Heap	$H ::= \emptyset \mid H, \ell \mapsto v$
Selection Path	$p ::= i \mid p.p$
Ivalues	$\mathcal{L} ::= 1 \mid \ell^\wedge \mid 1.p \mid \ell^\wedge.p$
Redex	$\mathcal{R} ::= - \mid \mathcal{R} e \mid v \mathcal{R} \mid \mathcal{L} := \mathcal{R}$ $\mid \text{if } \mathcal{R} \text{ then } e \text{ else } e \mid \mathcal{R}^\wedge$ $\mid \text{dup}(\mathcal{R}) \mid \text{dup}([\mathcal{R}]) \mid (\mathcal{R}, e)$ $\mid (v, \mathcal{R}) \mid \mathcal{R}.i \mid \text{let}^\kappa \underline{x} = \mathcal{R} \text{ in } e$

Grammar for Static Semantics:

Unf. Constraints	$u ::= \tau = \tau \mid \kappa = \varkappa \mid d$
Unf. Constraint Sets	$\mathcal{C} ::= \emptyset \mid \{\bar{u}\} \mid \mathcal{C} \cup \mathcal{C} \mid \mathcal{C} \cup \mathcal{C}$
Substitutions	$\theta ::= \langle \rangle \mid [\alpha \mapsto \tau] \mid [\kappa \mapsto \varkappa] \mid \theta \circ \theta$
Binding Environment	$\Gamma ::= \emptyset \mid \Gamma, x \mapsto \sigma$
Store Typing	$\Sigma ::= \emptyset \mid \Sigma, L \mapsto \tau$

Definition 1 (Mutability Normalization in Composite Types).

$\mathbb{M}(\tau) \mid \text{Immut}(\tau)$	$= \perp$
$\mathbb{M}(\alpha)$	$= \perp$
$\mathbb{M}(\Psi\alpha)$	$= \perp$
$\mathbb{M}(\Psi\text{bool})$	$= \langle \rangle$
$\mathbb{M}(\Psi\text{unit})$	$= \langle \rangle$
$\mathbb{M}(\Psi(\tau_1 \rightarrow \tau_2))$	$= \langle \rangle$
$\mathbb{M}(\Psi\uparrow\tau)$	$= \langle \rangle$
$\mathbb{M}(\Psi(\tau_1 \times \tau_2))$	$= \mathbb{M}(\tau_1) \circ \mathbb{M}(\tau_2)$
$\mathbb{M}(\alpha \downarrow \rho)$	$= [\alpha \mapsto \Psi\rho] \circ \mathbb{M}(\Psi\rho)$
$\mathbb{M}(\Psi\alpha \downarrow \rho)$	$= \langle \rangle$
$\mathbb{M}(\alpha \downarrow \rho)$	$= [\alpha \mapsto \Psi\beta] \mid \frac{\text{new}}{\text{new}} \beta$

Definition 2 (Normalization of Const Types).

$\mathbb{N}(\alpha)$	$= \alpha$
$\mathbb{N}([\alpha])$	$= [\alpha]$
$\mathbb{N}(\text{bool})$	$= \text{bool}$
$\mathbb{N}([\text{bool}])$	$= \text{bool}$
$\mathbb{N}(\text{unit})$	$= \text{unit}$
$\mathbb{N}([\text{unit}])$	$= \text{unit}$
$\mathbb{N}(\tau_1 \rightarrow \tau_2)$	$= \mathbb{N}(\tau_1) \rightarrow \mathbb{N}(\tau_2)$
$\mathbb{N}([\tau_1 \rightarrow \tau_2])$	$= \mathbb{N}(\tau_1) \rightarrow \mathbb{N}(\tau_2)$
$\mathbb{N}(\uparrow\tau)$	$= \uparrow\mathbb{N}(\tau)$
$\mathbb{N}([\uparrow\tau])$	$= \uparrow\mathbb{N}(\tau)$
$\mathbb{N}(\Psi\tau)$	$= \Psi\mathbb{N}(\tau)$
$\mathbb{N}([\Psi\tau])$	$= \mathbb{N}([\tau])$
$\mathbb{N}(\tau_1 \times \tau_2)$	$= \mathbb{N}(\tau_1) \times \mathbb{N}(\tau_2)$
$\mathbb{N}([\tau_1 \times \tau_2])$	$= [\mathbb{N}(\tau_1)] \times [\mathbb{N}(\tau_2)]$
$\mathbb{N}(\alpha \downarrow \rho)$	$= \alpha \downarrow \mathbb{N}(\Psi\rho)$
$\mathbb{N}([\alpha \downarrow \rho])$	$= [\alpha \downarrow \mathbb{N}(\Psi\rho)]$
$\mathbb{N}(\varsigma \downarrow \rho)$	$= \varsigma \downarrow \mathbb{N}(\rho)$
$\mathbb{N}([\varsigma \downarrow \rho])$	$= [\varsigma \downarrow \mathbb{N}(\rho)]$

Definition 3 (Notational convenience). We write $\tau_1 \overset{\nabla}{=} \tau_2$ as shorthand for $\nabla(\tau_1) = \nabla(\tau_2)$, $\tau_1 \overset{\forall}{=} \tau_2$ for $\forall(\tau_1) = \forall(\tau_2)$, and $\tau_1 \overset{\circ}{=} \tau_2$ for $\circ(\tau_1) = \circ(\tau_2)$

Definition 4 (Const-ness Requirement).

$\mathcal{P}(e, \tau)$	$= \text{true}$
$\mathcal{P}([e], [\tau])$	$= \text{true}$
$\mathcal{P}(\underline{e}, [\tau])$ (otherwise)	$= \text{false}$
$\mathbb{P}(e, \tau)$	$= \tau$
$\mathbb{P}([e], \tau)$	$= [\tau]$

Rule	Pre-conditions	Evaluation Step
E-Rval	$S(l) = v$	$S; H; l \Rightarrow S; H; v$
E-#	$S; H; e \Rightarrow S'; H'; e'$	$S; H; \mathcal{R}[e] \Rightarrow S'; H'; \mathcal{R}[e']$
E-App	$l \notin \text{dom}(S)$	$S; H; \lambda \underline{x}. e v \Rightarrow S; l \mapsto v; H; e[l/x]$
E-If	$b_1 = \text{true} \quad b_2 = \text{false}$	$S; H; \text{if } \bar{b}_i \text{ then } e_1 \text{ else } e_2 \Rightarrow S; H; e_i$
E-i		$S; H; (v_1, v_2).i \Rightarrow S; H; v_i$
E-Dup	$l \notin \text{dom}(H)$	$S; H; \text{dup}(\underline{v}) \Rightarrow S; H, l \mapsto v; l$
EL- \wedge #	$S; H; e \Rightarrow S'; H'; e'$	$S; H; e^\wedge \Rightarrow S'; H'; e'^\wedge$
E- \wedge	$H(l) = v$	$S; H; l^\wedge \Rightarrow S; H; v$
E-:=#	$S; H; l \Rightarrow S'; H'; l'$	$S; H; l := e \Rightarrow S'; H'; l' := e$
E-:=Stack		$S, l \mapsto v; H; l := v' \Rightarrow S, l \mapsto v'; H; ()$
E-:=Heap		$S; H, l \mapsto v; l^\wedge := v' \Rightarrow S; H, l \mapsto v'; ()$
E-:=S.p	$v'_{i_i} = v_{i_i} \quad S, l \mapsto v_i; H; l.p := v'_i$ $\Rightarrow S, l \mapsto v'_i; H; ()$	$S, l \mapsto (v_1, v_2); H; l.i.p := v'_i$ $\Rightarrow S, l \mapsto (v'_1, v'_2); H; ()$
E-:=H.p	$v'_{i_i} = v_{i_i} \quad S; H, l \mapsto v_i; l^\wedge.p := v'_i$ $\Rightarrow S; H, l \mapsto v'_i; ()$	$S; H, l \mapsto (v_1, v_2); l^\wedge.i.p := v'_i$ $\Rightarrow S; H, l \mapsto (v'_1, v'_2); ()$
E-Let-M	$l \notin \text{dom}(S)$	$S; H; \text{let}^\psi \underline{x} = v_1 \text{ in } e_2 \Rightarrow S, l \mapsto v_1; H; e_2[l/x]$
E-Let-P		$S; H; \text{let}^\forall \underline{x} = v_1 \text{ in } e_2 \Rightarrow S; H; e_2[v_1/x]$

Figure 1. Small Step Operational Semantics

τ	$\Delta(\tau)$	$\nabla(\tau)$	$\blacktriangle(\tau)$	$\blacktriangledown(\tau)$	$\diamond(\tau)$	$\mathcal{J}(\tau)$	$\{\tau\}$
α	$\Psi\alpha$	α	$\Psi\alpha$	α	α	α	$\{\alpha\}$
unit	Ψunit	unit	Ψunit	unit	unit	unit	\emptyset
bool	Ψbool	bool	Ψbool	bool	bool	bool	\emptyset
$\tau_1 \rightarrow \tau_2$	$\Psi(\tau_1 \rightarrow \tau_2)$	$\tau_1 \rightarrow \tau_2$	$\Psi(\tau_1 \rightarrow \tau_2)$	$\tau_1 \rightarrow \tau_2$	$\tau_1 \rightarrow \tau_2$	$\tau_1 \rightarrow \tau_2$	$\{\tau_1\} \cup \{\tau_2\}$
$\uparrow\tau$	$\Psi\uparrow\tau$	$\uparrow\tau$	$\Psi\uparrow\tau$	$\uparrow\tau$	$\uparrow\tau$	$\uparrow\mathcal{J}(\tau)$	$\{\tau\}$
$\Psi\rho$	$\Delta(\rho)$	$\nabla(\rho)$	$\blacktriangle(\rho)$	$\blacktriangledown(\rho)$	$\diamond(\rho)$	$\mathcal{J}(\rho)$	$\{\rho\}$
$\tau_1 \times \tau_2$	$\Psi(\Delta(\tau_1) \times \Delta(\tau_2))$	$\nabla(\tau_1) \times \nabla(\tau_2)$	$\Psi(\tau_1 \times \tau_2)$	$\tau_1 \times \tau_2$	$\diamond(\tau_1) \times \diamond(\tau_2)$	$\mathcal{J}(\tau_1) \times \mathcal{J}(\tau_2)$	$\{\tau_1\} \cup \{\tau_2\}$
$\alpha \downarrow \rho$	$\Delta(\rho)$	$\nabla(\rho)$	$\blacktriangle(\rho)$	$\blacktriangledown(\rho)$	$\diamond(\rho)$	$\mathcal{J}(\rho)$	$\{\alpha \downarrow \rho\} \cup \{\rho\}$
$\varsigma \downarrow \rho$	$\Delta(\rho)$	$\nabla(\rho)$	$\blacktriangle(\varsigma) \downarrow \rho$	$\blacktriangledown(\varsigma) \downarrow \rho$	$\diamond(\rho)$	$\mathcal{J}(\rho)$	$\{\varsigma \downarrow \rho\} \cup \{\rho\}$
$\lfloor \tau \rfloor$	$\Delta(\tau)$	$\lfloor \tau \rfloor$	$\Psi \lfloor \tau \rfloor$	$\lfloor \tau \rfloor$	$\diamond(\tau)$	$\mathcal{J}(\rho)$	$\{\tau\}$
τ	Mut(τ)	Immut(τ)	Const(τ)	$\theta(\tau)$			
α	false	false	false	τ if $[\alpha \mapsto \tau] \in \theta$, else α .			
unit	false	true	true	unit			
bool	false	true	true	bool			
$\tau_1 \rightarrow \tau_2$	false	true	true	$\theta(\tau_1) \rightarrow \theta(\tau_2)$			
$\uparrow\tau$	Mut(τ)	Immut(τ)	true	$\uparrow\theta(\tau)$			
$\Psi\rho$	true	false	false	$\Psi\theta(\rho)$			
$\tau_1 \times \tau_2$	Mut(τ_1) \vee Mut(τ_2)	Immut(τ_1) \wedge Immut(τ_2)	Const(τ_1) \wedge Const(τ_2)	$\theta(\tau_1) \times \theta(\tau_2)$			
$\alpha \downarrow \rho$	Mut($\blacktriangledown(\rho)$)	false	false	$\alpha' \downarrow \theta(\rho)$ if $\theta(\alpha) = \alpha'$			
				ρ' if $\theta(\alpha) = \rho' \neq \alpha'$			
				$\varsigma' \downarrow \theta(\rho)$ if $\theta(\varsigma) = \varsigma'$			
				ϱ if $\theta(\varsigma) = \varrho \neq \varsigma'$			
$\varsigma \downarrow \rho$	Mut(ς) \vee Mut($\nabla(\rho)$)	false	false	ϱ if $\theta(\varsigma) = \varrho \neq \varsigma'$			
$\lfloor \tau \rfloor$	false	Immut($\nabla(\tau)$)	true	$\lfloor \theta(\tau) \rfloor$			
τ	$\square(\tau)$	$\square(\tau)$	$\boxplus(\tau)$	$\mathbf{N}(\tau)$	$\mathfrak{M}(\tau)$		
α	false	false	false	α	\diamond		
unit	true	true	true	unit	\diamond		
bool	true	true	true	bool	\diamond		
$\tau_1 \rightarrow \tau_2$	true	true	true	$\mathbf{N}(\tau_1) \rightarrow \mathbf{N}(\tau_2)$	$\mathfrak{M}(\tau_1) \circ \mathfrak{M}(\tau_2)$		
$\uparrow\tau$	$\square(\tau)$	true	true	$\uparrow\mathbf{N}(\tau)$	$\mathfrak{M}(\tau)$		
$\Psi\rho$	$\square(\rho)$	$\square(\rho)$	$\boxplus(\rho)$	$\Psi\mathbf{N}(\rho)$	$\mathfrak{M}(\rho)$		
$\tau_1 \times \tau_2$	$\square(\tau_1) \wedge \square(\tau_2)$	$\square(\tau_1) \wedge \square(\tau_2)$	$\boxplus(\tau_1) \wedge \boxplus(\tau_2)$	$\mathbf{N}(\tau_1) \times \mathbf{N}(\tau_2)$	$\mathfrak{M}(\tau_1) \circ \mathfrak{M}(\tau_2)$		
$\alpha \downarrow \rho$	$\square(\rho)$	$\square(\rho)$	false	$\alpha \downarrow \mathbf{N}(\rho)$	$\mathfrak{M}(\rho)$		
$\varsigma \downarrow \rho$	$\square(\rho)$	$\square(\rho)$	false	$\varsigma \downarrow \mathbf{N}(\rho)$	$\mathfrak{M}(\rho) \circ [\alpha \mapsto \Delta(\rho)]$ if $\varsigma = \Psi\alpha$		
					and $\square(\rho)$; $\mathfrak{M}(\rho)$ otherwise		
$\lfloor \tau \rfloor$	$\square(\tau)$	$\square(\tau)$	true	$\nabla(\tau)$ if $\boxplus(\tau)$	$\mathfrak{M}(\tau)$		
				$\lfloor \tau \rfloor$ otherwise			

Meanings of Operators

Δ	Add mutability up to the ref/function boundary.
∇	Remove mutability up to the ref/function boundary.
\blacktriangle	Add top-most mutability.
\blacktriangledown	Remove top-most mutability.
\diamond	Remove mutability and const up to the ref/function boundary.
\mathcal{J}	Remove mutability deeply up to the function boundary.
$\{\} \}$	Constraint Collection
Mut	Is the type <i>observably</i> mutable?
Immut	Is the type <i>effectively</i> deeply immutable?
Const	Is the type <i>effectively</i> const?
$\theta(\langle \rangle)$	Substitution.
\square	Is the type <i>concretizable</i> ? (i.e. can be made invariable by a substitution <i>only</i> for mutability-variables?)
\square	Is the type <i>concretizable</i> up to a reference boundary?
\boxplus	Is this type concrete enough to drop an enclosing const?
\mathbf{N}	Partially normalized form of const types, when a const wrapper can be dropped (ex: $\lfloor \text{bool} \rfloor \equiv \text{bool}$)
\mathfrak{M}	Produce a substitution to normalize $\varsigma \downarrow \rho$ types. (ex: $\Psi\alpha \downarrow \text{bool} \equiv \Psi\text{bool}$)
\mathfrak{M}	Produce a substitution to propagate mutability inwards into composite types (or fail with an error).
\mathbf{N}	Normalized form of const types, where const only appears around variable or constrained types.

Figure 2. Operations and Predicates on Types

S-Refl $\frac{}{\tau \trianglelefteq: \tau}$	S-Trans $\frac{\tau_0 \trianglelefteq: \tau_1 \quad \tau_1 \trianglelefteq: \tau_2}{\tau_0 \trianglelefteq: \tau_2}$	S-Mut $\frac{\rho \trianglelefteq: \rho'}{\Psi\rho \trianglelefteq: \Psi\rho'}$	S-Pair $\frac{\tau_1 \trianglelefteq: \tau'_1 \quad \tau_2 \trianglelefteq: \tau'_2}{\tau_1 \times \tau_2 \trianglelefteq: \tau'_1 \times \tau'_2}$	S-MT1 $\frac{\nabla(\rho) \trianglelefteq: \tau}{\alpha \downarrow \rho \trianglelefteq: \tau}$	S-MT2 $\frac{\tau \trianglelefteq: \blacktriangle(\rho)}{\tau \trianglelefteq: \alpha \downarrow \rho}$
S-MF1 $\frac{\nabla(\rho) \trianglelefteq: \tau}{\alpha \downarrow \rho \trianglelefteq: \tau}$	S-MF2 $\frac{\alpha \downarrow \rho \trianglelefteq: \rho'}{\Psi\alpha \downarrow \rho \trianglelefteq: \Psi\rho'}$	S-MF3 $\frac{\tau \trianglelefteq: \Delta(\rho)}{\tau \trianglelefteq: \varsigma \downarrow \rho}$	S-Const1 $\frac{\tau \trianglelefteq: \nabla(\tau')}{\tau \trianglelefteq: [\tau']}$	S-Const2 $\frac{\nabla(\tau) \trianglelefteq: \tau'}{[\tau] \trianglelefteq: \tau'}$	

Figure 3. Copy Coercion Rules

T-Unit $\frac{}{\emptyset; \Gamma; \Sigma \vdash () : \text{unit}}$	T-Bool $\frac{}{\emptyset; \Gamma; \Sigma \vdash \text{b} : \text{bool}}$	T-Id $\frac{\Gamma(x) = \nabla\bar{\alpha}. \tau \setminus \mathcal{D} \quad \theta \Vdash \{\tau, \mathcal{D}\} \quad \text{dom}(\theta) = \{\bar{\alpha}\}}{\theta(\mathcal{D}); \Gamma; \Sigma \vdash x : \theta(\tau)}$
T-Hloc $\frac{\Sigma(\ell) = \tau}{\emptyset; \Gamma; \Sigma \vdash \ell : \uparrow\tau}$	T-Sloc $\frac{\Sigma(l) = \tau}{\emptyset; \Gamma; \Sigma \vdash l : \tau}$	T-Lambda $\frac{\mathcal{D}; \Gamma, x \mapsto \tau_1; \Sigma \vdash e : \tau_2 \quad \tau_1 \overset{\circ}{=} \tau'_1 \quad \tau_2 \overset{\circ}{=} \tau'_2 \quad \mathcal{P}(x, \tau_1)}{\mathcal{D}; \Gamma; \Sigma \vdash \lambda \underline{x}. e : \tau'_1 \rightarrow \tau'_2}$
T-App $\frac{\mathcal{D}_1; \Gamma; \Sigma \vdash e_1 \trianglelefteq: \tau_a \rightarrow \tau_r \quad \mathcal{D}_2; \Gamma; \Sigma \vdash e_2 \trianglelefteq: \nabla(\tau_a) \quad \Delta(\tau_r) \trianglelefteq: \tau}{\mathcal{D}_1 \cup \mathcal{D}_2; \Gamma; \Sigma \vdash e_1 e_2 : \tau}$		T-Set $\frac{\mathcal{D}_1; \Gamma; \Sigma \vdash l \trianglelefteq: \Psi\rho \quad \mathcal{D}_2; \Gamma; \Sigma \vdash e \trianglelefteq: \rho}{\mathcal{D}_1 \cup \mathcal{D}_2; \Gamma; \Sigma \vdash l := e : \text{unit}}$
T-If $\frac{\mathcal{D}_1; \Gamma; \Sigma \vdash e_1 \trianglelefteq: \text{bool} \quad \mathcal{D}_2; \Gamma; \Sigma \vdash e_2 \trianglelefteq: \tau \quad \mathcal{D}_3; \Gamma; \Sigma \vdash e_3 \trianglelefteq: \tau \quad \tau' \trianglelefteq: \tau}{\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3; \Gamma; \Sigma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau'}$		T-Deref $\frac{\mathcal{D}; \Gamma; \Sigma \vdash e \trianglelefteq: \uparrow\tau}{\mathcal{D}; \Gamma; \Sigma \vdash e^\wedge : \tau}$
T-Pair $\frac{\mathcal{D}_1; \Gamma; \Sigma \vdash e_1 \trianglelefteq: \tau_1 \quad \mathcal{D}_2; \Gamma; \Sigma \vdash e_2 \trianglelefteq: \tau_2 \quad \tau'_1 \trianglelefteq: \tau_1 \quad \tau'_2 \trianglelefteq: \tau_2}{\mathcal{D}_1 \cup \mathcal{D}_2; \Gamma; \Sigma \vdash (e_1, e_2) : \tau'_1 \times \tau'_2}$	T-Sel $\frac{\mathcal{D}; \Gamma; \Sigma \vdash e : \tau \quad \mathbb{N}(\tau) \overset{\nabla}{=} \tau_1 \times \tau_2}{\mathcal{D}; \Gamma; \Sigma \vdash e.i : \tau_i}$	
T-Dup $\frac{\mathcal{D}; \Gamma; \Sigma \vdash e \trianglelefteq: \tau \quad \tau' \trianglelefteq: \tau \quad \mathcal{P}(e, \tau')}{\mathcal{D}; \Gamma; \Sigma \vdash \text{dup}(e) : \uparrow\tau'}$	T-Let-M $\frac{\mathcal{D}_1; \Gamma; \Sigma \vdash e_1 \trianglelefteq: \tau_1 \quad \tau \trianglelefteq: \tau_1 \quad \mathcal{P}(x, \tau_1) \quad \mathcal{D}_2; \Gamma, x \mapsto \tau; \Sigma \vdash e_2 : \tau_2}{\mathcal{D}_1 \cup \mathcal{D}_2; \Gamma; \Sigma \vdash (\text{let}^\Psi \underline{x} = e_1 \text{ in } e_2) : \tau_2}$	
T-Let-MP $\frac{\mathcal{D}_1; \Gamma; \Sigma \vdash v \trianglelefteq: \tau_1 \quad \tau \trianglelefteq: \tau_1 \quad \mathcal{P}(x, \tau_1) \quad \mathcal{D} = \mathcal{D}_1 \cup \{\star_x^\times(\tau)\} \quad \{\bar{\alpha}\} = \text{ftv}(\tau, \mathcal{D}) \setminus \text{ftv}(\Gamma, \Sigma)}{\mathcal{D}_2; \Gamma, x \mapsto \nabla\bar{\alpha}. \tau \setminus \mathcal{D}; \Sigma \vdash e : \tau_2 \quad \frac{}{\text{new}} \bar{\beta}}{\mathcal{D}[\bar{\beta}/\bar{\alpha}] \cup \mathcal{D}_2; \Gamma; \Sigma \vdash (\text{let}^\times \underline{x} = v \text{ in } e) : \tau_2}$		

Figure 4. Declarative Type Rules

<p>I-Unit</p> $\frac{}{\Gamma; \Sigma \vdash () : \text{unit} \mid \emptyset}$	<p>I-Bool</p> $\frac{}{\Gamma; \Sigma \vdash b : \text{bool} \mid \emptyset}$	<p>I-Id</p> $\frac{\Gamma(x) = \forall \bar{\alpha}. \tau \setminus \mathcal{D} \quad \theta = \overline{[\alpha \mapsto \beta]} \quad \frac{}{\models_{\text{new}} \bar{\beta}}}{\Gamma; \Sigma \vdash x : \theta \langle \tau \rangle \mid \theta \langle \mathcal{D} \rangle}$
<p>I-Hloc</p> $\frac{\Sigma(\ell) = \tau}{\Gamma; \Sigma \vdash \ell : \uparrow \tau \mid \emptyset}$	<p>I-Sloc</p> $\frac{\Sigma(l) = \tau}{\Gamma; \Sigma \vdash l : \tau \mid \emptyset}$	<p>I-Lambda</p> $\frac{\Gamma, x \mapsto \mathbb{P}(x, \beta \downarrow \alpha); \Sigma \vdash e : \tau \mid \mathcal{C} \quad \frac{}{\models_{\text{new}} \alpha \beta \beta' \gamma \gamma' \delta}}{\Gamma; \Sigma \vdash \lambda \underline{x}. e : \beta' \downarrow \alpha \rightarrow \gamma' \downarrow \delta \mid \mathcal{C} \cup \{\tau = \gamma \downarrow \delta\}}$
<p>I-App</p> $\frac{\Gamma; \Sigma \vdash e_1 : \tau_1 \mid \mathcal{C}_1 \quad \Gamma; \Sigma \vdash e_2 : \tau_2 \mid \mathcal{C}_2 \quad \frac{}{\models_{\text{new}} \alpha \beta \beta' \gamma \gamma' \delta \varepsilon}}{\Gamma; \Sigma \vdash e_1 e_2 : \varepsilon \downarrow \gamma \mid \mathcal{C}_1 \cup \mathcal{C}_2 \cup \{\tau_1 = \alpha \downarrow (\beta' \downarrow \beta \rightarrow \gamma' \downarrow \gamma), \tau_2 = \delta \downarrow \beta\}}$	<p>I-Deref</p> $\frac{\Gamma; \Sigma \vdash e : \tau \mid \mathcal{C} \quad \frac{}{\models_{\text{new}} \alpha \beta}}{\Gamma; \Sigma \vdash e^\wedge : \alpha \mid \mathcal{C} \cup \{\tau = \beta \downarrow \uparrow \alpha\}}$	
<p>I-If</p> $\frac{\Gamma; \Sigma \vdash e_1 : \tau_1 \mid \mathcal{C}_1 \quad \Gamma; \Sigma \vdash e_2 : \tau_2 \mid \mathcal{C}_2 \quad \Gamma; \Sigma \vdash e_3 : \tau_3 \mid \mathcal{C}_3 \quad \frac{}{\models_{\text{new}} \alpha \beta \gamma \delta \varepsilon}}{\Gamma; \Sigma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \varepsilon \downarrow \gamma \mid \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \{\tau_1 = \alpha \downarrow \text{bool}, \tau_2 = \beta \downarrow \gamma, \tau_3 = \delta \downarrow \gamma\}}$		
<p>I-Set</p> $\frac{\Gamma; \Sigma \vdash l : \tau_1 \mid \mathcal{C}_1 \quad \Gamma; \Sigma \vdash e : \tau_2 \mid \mathcal{C}_2 \quad \frac{}{\models_{\text{new}} \alpha \beta \gamma}}{\Gamma; \Sigma \vdash l := e : \text{unit} \mid \mathcal{C}_1 \cup \mathcal{C}_2 \cup \{\tau_1 = (\Psi \alpha) \downarrow \beta, \tau_2 = \gamma \downarrow \beta\}}$	<p>I-Dup</p> $\frac{\Gamma; \Sigma \vdash e : \tau \mid \mathcal{C} \quad \tau' = \mathbb{P}(e, \alpha \downarrow \beta) \quad \frac{}{\models_{\text{new}} \alpha \beta \gamma}}{\Gamma; \Sigma \vdash \text{dup}(e) : \uparrow \tau' \mid \mathcal{C} \cup \{\tau = \gamma \downarrow \beta\}}$	
<p>I-Pair</p> $\frac{\Gamma; \Sigma \vdash e_1 : \tau_1 \mid \mathcal{C}_1 \quad \Gamma; \Sigma \vdash e_2 : \tau_2 \mid \mathcal{C}_2 \quad \frac{}{\models_{\text{new}} \alpha \alpha' \beta \beta' \gamma \delta}}{\Gamma; \Sigma \vdash (e_1, e_2) : \alpha \downarrow \gamma \times \beta \downarrow \delta \mid \mathcal{C}_1 \cup \mathcal{C}_2 \cup \{\tau_1 = \alpha' \downarrow \gamma, \tau_2 = \beta' \downarrow \delta\}}$	<p>I-Sel</p> $\frac{\Gamma; \Sigma \vdash e : \tau \mid \mathcal{C} \quad \tau_1 = \alpha \downarrow \beta \quad \tau_2 = \gamma \downarrow \delta \quad \frac{}{\models_{\text{new}} \alpha \beta \gamma \delta \varepsilon}}{\Gamma; \Sigma \vdash e.i : \tau_i \mid \mathcal{C} \cup \{\tau = \varepsilon \downarrow (\tau_1 \times \tau_2)\}}$	
<p>I-Let-Exp</p> $\frac{\Gamma; \Sigma \vdash e_1 : \tau_1 \mid \mathcal{C}_1 \quad e_1 \neq v \quad \Gamma, x \mapsto \mathbb{P}(x, \alpha \downarrow \beta); \Sigma \vdash e_2 : \tau_2 \mid \mathcal{C}_2 \quad \frac{}{\models_{\text{new}} \alpha \beta \gamma \kappa}}{\Gamma; \Sigma \vdash \text{let}^\kappa \underline{x} = e_1 \text{ in } e_2 : \tau_2 \mid \mathcal{C}_1 \cup \{\tau_1 = \gamma \downarrow \beta, \kappa = \psi\} \cup \mathcal{C}_2}$		
<p>I-Let-Val</p> $\frac{\Gamma; \Sigma \vdash v : \tau_1 \mid \mathcal{C}_1 \quad \mathcal{C}'_1 = \mathcal{C}_1 \cup \{\tau_1 = \gamma \downarrow \beta\} \quad \mathcal{U}(\mathcal{C}'_1) = (\mathcal{D}_u, \theta) \quad \mathcal{D} = \mathcal{D}_u \cup \{\star_x^\kappa(\tau)\} \quad \tau = \mathbb{P}(\underline{x}, \theta \langle \delta \downarrow \beta \rangle) \quad \{\bar{\alpha}\} = \text{ftv}(\tau, \mathcal{D}) \setminus \text{ftv}(\theta \langle \Gamma \rangle, \theta \langle \Sigma \rangle) \quad \Gamma, x \mapsto \forall \bar{\alpha}. \tau \setminus \mathcal{D}; \Sigma \vdash e : \tau_2 \mid \mathcal{C}_2 \quad \frac{}{\models_{\text{new}} \beta \gamma \delta \bar{\varepsilon} \kappa}}{\Gamma; \Sigma \vdash \text{let}^\kappa \underline{x} = v \text{ in } e : \tau_2 \mid \mathcal{C}'_1[\bar{\varepsilon}/\bar{\alpha}] \cup \mathcal{C}_2}$		

Figure 5. Type Inference Algorithm

U-Empty	$\mathcal{U}(\emptyset)$	=	$(\emptyset, \langle \rangle)$
U-Refl	$\mathcal{U}(\{\tau = \tau\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C})$
U-Sym	$\mathcal{U}(\{\tau_1 = \tau_2\} \cup \mathcal{C})$	=	$\mathcal{U}(\{\tau_2 = \tau_1\} \cup \mathcal{C})$
U-Var	$\mathcal{U}(\{\alpha = \tau\} \cup \mathcal{C}) \mid \alpha \notin \tau$	=	$(\mathcal{D}, \theta_a \circ \theta_u)$ where $\theta_a = [\alpha \mapsto \tau]$ and $\mathcal{U}(\mathbb{N}(\theta_a \langle \mathcal{C} \rangle)) = (\mathcal{D}, \theta_u)$
U-Fn	$\mathcal{U}(\{\tau_a \rightarrow \tau_r = \tau'_a \rightarrow \tau'_r\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\tau_a = \tau'_a, \tau_r = \tau'_r\})$
U-Ref	$\mathcal{U}(\{\uparrow\tau_1 = \uparrow\tau_2\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\tau_1 = \tau_2\})$
U-Mut	$\mathcal{U}(\{\Psi\rho_1 = \Psi\rho_2\} \cup \mathcal{C})$	=	$(\mathcal{D}, \theta_u \circ \theta_m)$ where $\mathbb{M}(\theta_u \langle \tau_1 \rangle) = \theta_m$ and $\mathcal{U}(\mathcal{C} \cup \{\rho_1 = \rho_2\}) = (\mathcal{D}, \theta_u)$
U-Pair	$\mathcal{U}(\{\tau_1 \times \tau_2 = \tau'_1 \times \tau'_2\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\tau_1 = \tau'_1, \tau_2 = \tau'_2\})$
U-Const1	$\mathcal{U}(\{\lfloor \tau_1 \rfloor = \lfloor \tau_2 \rfloor\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\tau_1 \overset{\circ}{=} \tau_2\})$
U-Const2	$\mathcal{U}(\{\lceil \tau_1 \rceil = \tau_2\} \cup \mathcal{C})$ where $\mathbb{N}(\tau_1) \neq \tau_1$	=	$\mathcal{U}(\mathcal{C} \cup \{\mathbb{N}(\tau_1) = \tau_2\})$
U-Ct1	$\mathcal{U}(\{\alpha \downarrow \rho_1 = \beta \downarrow \rho_2\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\rho_1 \overset{\nabla}{=} \rho_2, \alpha = \beta\})$
U-Ct2	$\mathcal{U}(\{\alpha \downarrow \rho = \rho'\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\rho \overset{\nabla}{=} \rho', \alpha = \rho'\})$
U-Ct3	$\mathcal{U}(\{\alpha \downarrow \rho_1 = \varsigma \downarrow \rho_2\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\rho_1 \overset{\circ}{=} \rho_2, \alpha = \varsigma\})$
U-Ct4	$\mathcal{U}(\{\Psi\alpha \downarrow \rho_1 = \Psi\beta \downarrow \rho_2\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\rho_1 \overset{\circ}{=} \rho_2, \alpha = \beta\})$
U-Ct5	$\mathcal{U}(\{\varsigma \downarrow \rho = \varrho\} \cup \mathcal{C})$	=	$\mathcal{U}(\mathcal{C} \cup \{\rho \overset{\circ}{=} \varrho, \varsigma = \varrho\})$
U-K	$\mathcal{U}(\{\kappa = \varkappa\} \cup \mathcal{C})$	=	$(\mathcal{D}, \theta_k \circ \theta_u)$ where $\theta_k = [\kappa \mapsto \varkappa]$ and $\mathcal{U}(\theta_k \langle \mathcal{C} \rangle) = (\mathcal{D}, \theta_u)$
U-Om1	$\mathcal{U}(\{\star_x^\psi(\tau_1), \star_x^\psi(\tau_2)\} \cup \mathcal{C})$	=	$(\mathcal{D} \cup \theta\{\star_x^\psi(\tau_1), \star_x^\psi(\tau_2)\}, \theta)$ where $\mathcal{U}(\mathcal{C} \cup \{\tau_1 = \tau_2\}) = (\mathcal{D}, \theta)$
U-Op1	$\mathcal{U}(\{\star_x^\forall(\tau)\} \cup \mathcal{C}) \mid \Box(\tau)$	=	$(\mathcal{D} \cup \theta\{\star_x^\forall(\tau)\}, \theta)$ where $\mathcal{U}(\mathcal{C} \cup \{\tau = \mathcal{I}(\tau)\}) = (\mathcal{D}, \theta)$
U-Om2	$\mathcal{U}(\{\star_x^\kappa(\tau)\} \cup \mathcal{C}) \mid \text{Mut}(\tau)$	=	$(\mathcal{D}, \theta_k \circ \theta_u)$ where $\theta_k = [\kappa \mapsto \psi]$ and $\mathcal{U}(\theta_k \langle \{\star_x^\kappa(\tau)\} \cup \mathcal{C} \rangle) = (\mathcal{D}, \theta_u)$
U-Op2	$\mathcal{U}(\{\star_x^\kappa(\tau_1), \star_x^\kappa(\tau_2)\} \cup \mathcal{C})$ where $\mathcal{U}(\{\tau_1 = \tau_2\} \cup \mathcal{C}) = \perp$	=	$(\mathcal{D}, \theta_k \circ \theta_u)$ where $\theta_k = [\kappa \mapsto \forall]$ and $\mathcal{U}(\theta_k \langle \{\star_x^\kappa(\tau_1), \star_x^\kappa(\tau_2)\} \cup \mathcal{C} \rangle) = (\mathcal{D}, \theta_u)$
U-Error	$\mathcal{U}(c \cup \mathcal{C}) \mid c \notin \mathcal{C}_v \cup \mathcal{C}_s \cup \mathcal{C}_p$	=	\perp
	$\mathcal{C}_v = \forall \alpha, \varsigma, \rho, \tau, \tau' \mid \alpha \notin \tau'. \{\tau = \tau, \alpha = \tau', \tau' = \alpha, \alpha \downarrow \rho = \tau, \tau = \alpha \downarrow \rho, \varsigma \downarrow \rho = \tau, \tau = \varsigma \downarrow \rho\}$		
	$\mathcal{C}_s = \forall \rho, \rho', \tau, \tau', \tau_1, \tau'_1. \{\tau \rightarrow \tau_1 = \tau' \rightarrow \tau'_1, \uparrow\tau = \uparrow\tau', \Psi\rho = \Psi\rho', \lfloor \tau \rfloor = \lfloor \tau' \rfloor\}$		
	$\mathcal{C}_p = \forall x, \kappa, \varkappa, \tau, \tau' \mid \neg \text{Mut}(\tau'). \{\kappa = \varkappa, \star_x^\psi(\tau), \star_x^\forall(\tau'), \star_x^\kappa(\tau)\}$		

Figure 6. Unification Algorithm