An impure language: Immutable References

By Amal Ahmed, PhD thesis (Section 2.2), Princeton University, 2004
Presenters: Patrick van Beurden & Razvan Potcoveanu
04-12-2023

Syntax of λ^I

Syntax The syntax of λ^I terms is given by the following grammar.

$$
\begin{array}{lcll}
\textit{Values} & v & ::= & \ell \mid \texttt{true} \mid \texttt{false} \mid \lambda x.e \\
\textit{Expressions} & e & ::= & x \mid v \mid (e_1\, e_2) \mid \texttt{new}(e) \mid !\,e
\end{array}
$$

- A term *e* is a value if it is a location *l*, a boolean constant, or an abstraction with no free variables.
- *new* is a term for allocating a new cell on the heap, and term *!* reads the contents of an allocated cell.

Operational semantics of *new* and *!*

$$\frac{(S, e) \longmapsto_I (S', e')}{(S, \mathtt{new}(e)) \longmapsto_I (S', \mathtt{new}(e'))}\ (\mathtt{IO\text{-}new1})$$

$$\frac{(S, e) \longmapsto_I (S', e')}{(S, !\,e) \longmapsto_I (S', !\,e')}\ (\mathtt{IO\text{-}deref1})$$

$$\frac{\ell \notin \mathrm{dom}(S)}{(S, \mathtt{new}(v)) \longmapsto_I (S\,[\ell \mapsto v], \ell)}\ (\mathtt{IO\text{-}new2})$$

$$\frac{\ell \in \mathrm{dom}(S)}{(S, !\,\ell) \longmapsto_I (S, S(\ell))}\ (\mathtt{IO\text{-}deref2})$$

Note: In a language with side effects, the order in which terms are evaluated is important.

Safety

Definition 2.12 (Safe)
A state (S, e) *is* safe *if whenever* (S, e) *evaluates to a state* (S', e'), *either* e' *is a value or another step is possible.*

$$\begin{aligned} \mathrm{safe}(S, e) \quad \stackrel{\mathrm{def}}{=} \quad & \forall S', e'.\ (S, e) \longmapsto_I^* (S', e') \\ & \implies (\mathrm{val}(e') \ \vee\ \exists S'', e''.\ (S', e') \longmapsto_I (S'', e'')) \end{aligned}$$

Type definitions of λ^{I}

$$\begin{aligned}
\mathsf{bool} &\stackrel{\text{def}}{=} \\
\mathsf{box}\ \tau &\stackrel{\text{def}}{=} \{\, \langle S, \ell \,\rangle \mid \ell \in \mathrm{dom}(S) \ \wedge\ \langle S, S(\ell)\rangle \in \tau \,\} \\
\tau_1 \longrightarrow \tau_2 &\stackrel{\text{def}}{=}
\end{aligned}$$

Type definitions of λ^{l}

$$
\begin{aligned}
\mathsf{bool} &\stackrel{\text{def}}{=} \\
\mathsf{box}\ \tau &\stackrel{\text{def}}{=} \{\, \langle S, \ell \,\rangle \mid \ell \in \mathrm{dom}(S) \ \wedge \ \langle S, S(\ell)\rangle \in \tau \,\} \\
\tau_1 \longrightarrow \tau_2 &\stackrel{\text{def}}{=}
\end{aligned}
$$

	% S_0
$\texttt{let}\ x_1 = \texttt{new(true)}\ \texttt{in}$	% $S_1 = S_0\,[\ell_1 \mapsto \texttt{true}],\ \ \ell_1 \notin \mathrm{dom}(S_0),\ \ x_1 \mapsto \ell_1$
$\texttt{let}\ x_2 = \ \ldots\ \ \texttt{in}$	% $S_2 = \ \ldots,\ x_1 \mapsto \ell_1$
$\vdots$	
$\texttt{let}\ x_n = \ \ldots\ \ \texttt{in}$	% $S_n = \ \ldots,\ x_1 \mapsto \ell_1$
$!\,x_1$	% $S_{n+1} = S_n$

Type definitions of λ^{l}

$$
\begin{aligned}
\text{bool} &\stackrel{\text{def}}{=} \\
\text{box } \tau &\stackrel{\text{def}}{=} \{ \langle S, \ell \rangle \mid \ell \in \mathrm{dom}(S) \ \wedge \ \langle S, S(\ell) \rangle \in \tau \} \\
\tau_1 \longrightarrow \tau_2 &\stackrel{\text{def}}{=}
\end{aligned}
$$

Definition 2.13 (Store Extension)
A valid store extension *is defined as follows:*

$$S \sqsubseteq S' \stackrel{\text{def}}{=} \forall \ell.\ \ell \in \mathrm{dom}(S) \implies (\ell \in \mathrm{dom}(S') \ \wedge \ S(\ell) = S'(\ell))$$

Type definitions of λ^{I}

$$\begin{aligned}
\mathsf{bool} &\stackrel{\text{def}}{=} \{\langle S, v\rangle \mid v = \mathtt{true} \ \vee\ v = \mathtt{false}\} \\
\mathsf{box}\ \tau &\stackrel{\text{def}}{=} \\
\tau_1 \rightarrow \tau_2 &\stackrel{\text{def}}{=}
\end{aligned}$$

Definition 2.14 (Type)
A type *is a set τ of tuples of the form $\langle S, v\rangle$, where S is a store and v is a value, such that if $\langle S, v\rangle \in \tau$ and $S \sqsubseteq S'$ then $\langle S', v\rangle \in \tau$; that is,*

$$\mathrm{type}(\tau) \stackrel{\text{def}}{=} \forall S, S', v.\ (\langle S, v\rangle \in \tau \ \wedge\ S \sqsubseteq S') \implies \langle S', v\rangle \in \tau$$

Type definitions of λ^{I}

$$\mathsf{bool} \stackrel{\text{def}}{=}$$

$$\mathsf{box}\ \tau \stackrel{\text{def}}{=}$$

$$\tau_1 \longrightarrow \tau_2 \stackrel{\text{def}}{=}$$

Definition 2.15 (Expr : Type)

For any closed expression e *and type* τ *I write* $e :_S \tau$ *if whenever* $(S, e) \longmapsto_I^* (S', e')$ *and* (S', e') *is irreducible, then* $\langle S, e' \rangle \in \tau$*; that is,*

$$\begin{aligned} e :_S \tau \stackrel{\text{def}}{=}\ & \forall S', e'.\ ((S, e) \longmapsto_I^* (S', e') \ \wedge\ \mathrm{irred}(S', e')) \\ & \Longrightarrow\ S \sqsubseteq S' \ \wedge\ \langle S', e' \rangle \in \tau \end{aligned}$$

Arrow

Type definitions of λ^{I}

$$
\begin{aligned}
\mathsf{bool} &\stackrel{\text{def}}{=} \\
\mathsf{box}\ \tau &\stackrel{\text{def}}{=} \\
\tau_1 \rightarrow \tau_2 &\stackrel{\text{def}}{=} \{\, \langle S, \lambda x.e \rangle \mid \forall v, S'.\ (S \sqsubseteq S' \ \wedge\ \langle S', v \rangle \in \tau_1) \implies e[v/x] :_{S'} \tau_2 \,\}
\end{aligned}
$$

Definition 2.15 (Expr : Type)

For any closed expression e and type τ I write $e :_S \tau$ if whenever $(S, e) \longmapsto_I^* (S', e')$ and (S', e') is irreducible, then $\langle S, e' \rangle \in \tau$; that is,

$$
\begin{aligned}
e :_S \tau \stackrel{\text{def}}{=} \ & \forall S', e'.\ ((S, e) \longmapsto_I^* (S', e') \ \wedge\ \mathrm{irred}(S', e')) \\
& \implies S \sqsubseteq S' \ \wedge\ \langle S', e' \rangle \in \tau
\end{aligned}
$$

A modal interpresentation - Possible Worlds

$$
\begin{aligned}
\mathsf{bool} &\stackrel{\text{def}}{=} \{\, \langle S, v\rangle \mid v = \mathtt{true} \;\vee\; v = \mathtt{false} \,\} \\
\mathsf{box}\ \tau &\stackrel{\text{def}}{=} \{\, \langle S, \ell\rangle \mid \ell \in \mathrm{dom}(S) \;\wedge\; \langle S, S(\ell)\rangle \in \tau \,\} \\
\tau_1 \rightarrow \tau_2 &\stackrel{\text{def}}{=} \{\, \langle S, \lambda x.e\rangle \mid \forall v, S'.\ (S \sqsubseteq S' \;\wedge\; \langle S', v\rangle \in \tau_1) \implies e[v/x] :_{S'} \tau_2 \,\}
\end{aligned}
$$

To define:

- A set W of worlds $W = Loc \stackrel{\text{fin}}{\rightarrow} Val$
- An accessibility relation $S \sqsubseteq S'$
- A labelling function $L(S) = \{\, (\ell, v) \mid S(\ell) = v \,\}$
- Properties Acc should satisfy (reflexivity, transitivity) $\sqsubseteq$

Validity of types

$$\text{type}(\tau) \stackrel{\text{def}}{=} \forall S, S', v.\ ((\langle S, v\rangle \in \tau \ \wedge\ S \sqsubseteq S') \implies \langle S', v\rangle \in \tau$$

$$\textsf{bool} \stackrel{\text{def}}{=} \{\, \langle S, v\rangle \mid v = \texttt{true} \ \vee\ v = \texttt{false}\,\}$$

Lemma 2.16 (Store Extension Reflexive)
$S \sqsubseteq S$.

Lemma 2.17 (Store Extension Transitive)
If $S_1 \sqsubseteq S_2$ *and* $S_2 \sqsubseteq S_3$ *then* $S_1 \sqsubseteq S_3$.

Validity of types

$$\text{type}(\tau) \stackrel{\text{def}}{=} \forall S, S', v.\ (\langle S, v\rangle \in \tau \ \wedge\ S \sqsubseteq S') \implies \langle S', v\rangle \in \tau$$

$$\tau_1 \to \tau_2 \stackrel{\text{def}}{=} \{\langle S, \lambda x.e\rangle \mid \forall v, S'.\ (S \sqsubseteq S' \ \wedge\ \langle S', v\rangle \in \tau_1) \implies e[v/x] :_{S'} \tau_2\}$$

Lemma 2.18 (Type $\tau_1 \to \tau_2$)
If τ_1 and τ_2 are types then $\tau_1 \to \tau_2$ is also a type.

1. Suppose $\langle S, v\rangle \in \tau_1 \to \tau_2$ and $S \sqsubseteq S'$ (with v of the form $\lambda x.e$)
2. We have to show that $\langle S', \lambda x.e\rangle \in \tau_1 \to \tau_2$
3. Suppose $S' \sqsubseteq S''$ and $\langle S'', v_1\rangle \in \tau_1$
4. By definition of →, we now need to show: $e[v_1/x] :_{S''} \tau_2$
5. Transitivity lemma gives us $S \sqsubseteq S''$
6. By definition of →, and $\langle S, \lambda x.e\rangle \in \tau_1 \to \tau_2$ with 5 and $\langle S'', v_1\rangle \in \tau_1$ implies $e[v_1/x] :_{S''} \tau_2$

closed under store extension

Validity of types

$$\text{type}(\tau) \stackrel{\text{def}}{=} \forall S, S', v.\ ((\langle S, v\rangle \in \tau \ \wedge\ S \sqsubseteq S') \implies \langle S', v\rangle \in \tau$$

$$\text{box } \tau \stackrel{\text{def}}{=} \{ \langle S, \ell \rangle \mid \ell \in \text{dom}(S) \ \wedge\ \langle S, S(\ell)\rangle \in \tau \}$$

Lemma 2.19

If τ is a type, then box τ *is a type.*

1. Suppose $\langle S, v\rangle \in$ box τ and $S \sqsubseteq S'$ (with v some store location ℓ)
2. We have to show that $\langle S', v\rangle \in$ box τ
3. From $\langle S, \ell \rangle \in$ box τ we have that:
   a. $\ell \in \text{dom}(S)$
   b. $\langle S, S(\ell)\rangle \in \tau$
4. Since τ is a type, we have $\langle S', S(\ell)\rangle \in \tau$
5. From 3a and $S \sqsubseteq S'$ it follows that $\ell \in \text{dom}(S')$ and $S(\ell) = S'(\ell)$
6. Hence, we have $\langle S', S'(\ell)\rangle \in \tau$, so by definition of *box* we have $\langle S', \ell \rangle \in$ box τ

Typing rules

$$\frac{}{\Gamma \vDash_I x : \Gamma(x)} \quad (\texttt{I-var})$$

$$\frac{}{\Gamma \vDash_I \texttt{true} : \mathsf{bool}} \quad (\texttt{I-true}) \qquad \frac{}{\Gamma \vDash_I \texttt{false} : \mathsf{bool}} \quad (\texttt{I-false})$$

$$\frac{\Gamma\,[x \mapsto \tau_1] \vDash_I e : \tau_2}{\Gamma \vDash_I \lambda x.e : \tau_1 \to \tau_2} \quad (\texttt{I-abs})$$

$$\frac{\Gamma \vDash_I e_1 : \tau_1 \to \tau_2 \qquad \Gamma \vDash_I e_2 : \tau_1}{\Gamma \vDash_I (e_1\, e_2) : \tau_2} \quad (\texttt{I-app})$$

$$\frac{\Gamma \vDash_I e : \tau}{\Gamma \vDash_I \mathsf{new}(e) : \mathsf{box}\ \tau} \quad (\texttt{I-new}) \qquad \frac{\Gamma \vDash_I e : \mathsf{box}\ \tau}{\Gamma \vDash_I\, !e : \tau} \quad (\texttt{I-deref})$$

e

Semantics of Judgement

For any type of environment Γ and a value environment σ I write $\sigma :_S \Gamma$ if for all variables $x \in \mathrm{dom}(\Gamma)$ we have $\sigma(x) :_S \Gamma(x)$; that is,

$$\sigma :_S \Gamma \stackrel{\text{def}}{=} \forall x \in \mathrm{dom}(\Gamma).\ \sigma(x) :_S \Gamma(x)$$

I write $\Gamma \vDash_I e : \tau$ *iff* $FV(e) \subseteq \mathrm{dom}(\Gamma)$ and $\forall \sigma, S.\ \sigma :_S \Gamma \implies \sigma(e) :_S \tau$ where $\sigma(e)$ is the result of replacing the free variables in e with their values under σ.

I write $\vDash_I e : \tau$ to mean $\Gamma_0 \vDash_I e : \tau$ for the empty environment Γ_0.

Safety

Theorem 2.21 (Safety)
If $\vDash_I e : \tau$, τ is a type, and S is a store, then (S, e) is safe.

PROOF:

1. Suppose $(S, e) \longmapsto_I^* (S', e')$
2. If (S', e') is not irreducible, then $\exists\,(S'', e'')$ such that $(S', e') \longmapsto_I (S'', e'')$
3. Otherwise, (S', e') is irreducible and we must prove that e' is a value
4. $\vDash_I e : \tau$ => $\Gamma_0 \vDash_I e : \tau$ => e is closed
5. Choose the empty value environment σ_0 and store S
6. By the definition of $\vDash_I$ we have $\sigma_0 :_S \Gamma_0 \implies \sigma_0(e) :_S \tau$
7. Since $(S, e) \longmapsto_I^* (S', e')$ and $\mathrm{irred}(S', e') \Longrightarrow S \sqsubseteq S'$ and $\langle S', e' \rangle \in \tau$
8. τ is a type $\Longrightarrow \mathrm{val}(e')$

Validity of I-New

Lemma 2.24 (Closed New)
If e is a closed term and τ is a type such that $e :_S \tau$ then $\texttt{new}(e) :_S \mathsf{box}\ \tau$.

Proof:

Suppose $(S, \texttt{new}(e)) \longmapsto_I^* (S', \ell)$ where $\mathrm{irred}(S', \ell)$

$S \sqsubseteq S'$ and $\langle S', \ell \rangle \in \mathsf{box}\ \tau$.

Part 1

1. $(S, e) \longmapsto_I^* (S_1, e_1)$ and $\mathrm{irred}(S_1, e_1)$
2. From the premise $e :_S \tau$ we have $S \sqsubseteq S_1$ and $\langle S_1, e_1 \rangle \in \tau$

Part 2

1. $(S_1, \texttt{new}(e_1)) \longmapsto_I (S_1[\ell \mapsto e_1], \ell)$
2. l is a value => $\mathrm{irred}(S_1[\ell \mapsto e_1], \ell)$
3. $S_1 \sqsubseteq S'$ then, by the transitivity of $\sqsubseteq$ => $S \sqsubseteq S'$

$\langle S_1, e_1 \rangle \in \tau$ and $S_1 \sqsubseteq S'$ => $\langle S', e_1 \rangle \in \tau$; $S'(\ell) = e_1$ => $\langle S', S'(\ell) \rangle \in \tau$ which together with $\ell \in S'$ => $\langle S', \ell \rangle \in \mathsf{box}$

Validity of I-New

Theorem 2.25 (New)
If Γ is a type environment, e is a (possibly open) term, and τ is a type such that $\Gamma \models_I e : \tau$, then $\Gamma \models_I \mathtt{new}(e) : \mathsf{box}\ \tau$.

Proof:

1. Suppose $\sigma :_S \Gamma$
2. From the premise $\Gamma \models_I e : \tau$ we have $\sigma(e) :_S \tau$.
3. Since $\sigma(e)$ is a closed term, the result now follows from Lemma 2.24

Validity of I-Deref

Lemma 2.26 (Closed Deref)
If e is a closed term and τ is a type such that $e :_S \mathsf{box}\ \tau$ then $!e :_S \tau$.

Proof:

1. Suppose $(S, !e) \longmapsto_I^* (S', e')$ such that $\mathrm{irred}(S', e')$
2. $(S, e) \longmapsto_I^* (S', \ell)$ and $\mathrm{irred}(S', \ell)$
3. $e :_S \mathsf{box}\ \tau$ => $S \sqsubseteq S'$ and $\langle S', \ell \rangle \in \mathsf{box}\ \tau$
4. From the definition of $\mathsf{box}\ \tau$ we have $\ell \in \mathrm{dom}(S')$ and $\langle S', S'(\ell) \rangle \in \tau$
5. From the operational semantics we have $(S', !\ell) \longmapsto_I (S', e')$ where $e' = S'(\ell)$
6. Hence, $\langle S', e' \rangle \in \tau$

Theorem 2.27 (Deref)
If Γ is a type environment, e is a (possibly open) term, and τ is a type such that $\Gamma \vDash_I e : \mathsf{box}\ \tau$, then $\Gamma \vDash_I !e : \tau$.

Kripke Logical Relations

There is a correspondence between Kripke logical relations and the possible-worlds model developed for λ^I .

A Kripke logical relation over a set A is defined as follows:

Suppose that we have a set of worlds W , an accessibility relation $Acc \subseteq W \times W$, and a family $\{i^{\tau}_{w\,w'}\}$, of transition functions where w and w' are worlds such that $Acc(w, w')$. A Kripke logical relation is a family $\mathcal{R} = \{R^{\tau}_{w}\}$ of relations R^{τ}_{w} indexed by types τ and worlds $w \in W$, satisfying the condition:

$$R^{\tau}_{w}(a) \text{ implies } R^{\tau}_{w'}(i^{\tau}_{w\,w'}(a)) \text{ for all } w' \text{ such that } Acc(w, w')$$