Explicit CN Soundness Proof

Dhruv Makwana

June 24, 2021

1 Weakening

If $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$ and $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash J$ then $\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}' \vdash J$.

Proof sketch: Induction over the typing judgements.

Assume: 1. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$.
2. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash J$.

Prove: $\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}' \vdash J$.

2 Substitution

2.1 Weakening for Substitution

Weakening for substitution: as above, but with $J = (\sigma) : (\mathcal{C}''; \mathcal{L}''; \Phi''; \mathcal{R}'')$.

Proof sketch: Induction over the substitution.

Assume: 1. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \sqsubseteq \mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'$.
2. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}''; \mathcal{L}''; \Phi''; \mathcal{R}'')$.

Prove: $\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}' \vdash (\sigma){:}(\mathcal{C}''; \mathcal{L}''; \Phi''; \mathcal{R}'')$.

2.2 Substitution Lemma

If $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$ and $\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}' \vdash J$ then $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \sigma(J)$.

Proof sketch: Induction over the typing judgements.

Assume: 1. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$.
2. $\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}' \vdash J$.

Prove: $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \sigma(J)$.

$\langle 1 \rangle$1. Case: Ty_PVal_Var.
$\mathcal{C}'; \mathcal{L}'; \Phi' \vdash x \Rightarrow \beta$

$\langle 2 \rangle$1. Have $x{:}\beta \in \mathcal{C}'$ (or $x{:}\beta \in \mathcal{L}'$).

$\langle 2 \rangle$2. So $\exists \mathit{pval}.\ \mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta$ by Ty_Subs_Cons_{Comp,Log}.

$\langle 2 \rangle$3. Since $\mathit{pval} = \sigma(x)$, we are done.

⟨1⟩2. CASE: TY_TPE_LET.
$\mathcal{C}'; \mathcal{L}'; \Phi' \vdash \mathtt{let}\, \mathit{ident\_or\_pattern} = \mathit{pexpr}\, \mathtt{in}\, \mathit{tpexpr} \Leftarrow y_2{:}\beta_2.\ \mathit{term}_2$.

⟨2⟩1. By induction,
1. $\mathcal{C}; \mathcal{L}; \Phi \vdash \sigma(\mathit{pexpr}) \Rightarrow y_1{:}\beta.\ \sigma(\mathit{term}_1)$
2. $\mathcal{C}, \mathcal{C}_1; \mathcal{L}, y_1{:}\beta; \Phi, \mathit{term}_1, \Phi' \vdash \sigma(\mathit{tpexpr}) \Leftarrow y_2{:}\beta.\ \sigma(\mathit{term}_2)$.

⟨2⟩2. $\mathcal{C}; \mathcal{L}; \Phi \vdash \sigma(\mathtt{let}\, \mathit{ident\_or\_pattern} = \mathit{pexpr}\, \mathtt{in}\, \mathit{tpexpr}) \Leftarrow y_2{:}\beta_2.\ \sigma(\mathit{term}_2)$ as required.

⟨1⟩3. CASE: TY_TVAL_LOG.
$\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}' \vdash \mathtt{done}\, \mathit{pval},\ \overline{\mathit{spine\_elem}_i}^{\,i} \Leftarrow \exists\, y{:}\beta.\ \mathit{ret}$.

⟨2⟩1. By inversion and then induction,
1. $\mathcal{C}; \mathcal{L}; \Phi \vdash \sigma(\mathit{pval}) \Rightarrow \beta$
2. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \sigma(\mathtt{done}\, \overline{\mathit{spine\_elem}_i}^{\,i}) \Leftarrow \sigma(\mathit{pval}/y, \cdot(\mathit{ret}))$.

⟨2⟩2. Therefore $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \sigma(\mathtt{done}\, \mathit{pval},\ \overline{\mathit{spine\_elem}_i}^{\,i}) \Leftarrow \exists\, y{:}\beta.\ \sigma(\mathit{ret})$.

⟨1⟩4. CASE: TY_SPINE_RES.
$\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}'_1, \mathcal{R}_2 \vdash x = \mathit{res\_term},\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{res} \multimap \mathit{arg} \gg \mathit{res\_term}/x, \psi; \mathit{ret}$

⟨2⟩1. By inversion and then induction,
1. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1 \vdash \sigma(\mathit{res\_term}) \Leftarrow \sigma(\mathit{res})$.
2. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_2 \vdash \ \overline{x_i = \sigma(\mathit{spine\_elem}_i)}^{\,i} :: \sigma(\mathit{res}) \multimap \sigma(\mathit{arg}) \gg \sigma(\psi); \sigma(\mathit{ret})$.

⟨2⟩2. Hence $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R}_2 \vdash x = \sigma(\mathit{res\_term}),\ \overline{x_i = \sigma(\mathit{spine\_elem}_i)}^{\,i} :: \sigma(\mathit{res} \multimap \mathit{arg}) \gg \sigma(\mathit{res\_term}/x, \psi); \sigma(\mathit{ret})$ as required.

2.3 Identity Extension

If $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$ then $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R} \vdash (\sigma, \mathrm{id}){:}(\mathcal{C}, \mathcal{C}'; \mathcal{L}, \mathcal{L}'; \Phi, \Phi'; \mathcal{R}_1, \mathcal{R}')$.

PROOF SKETCH: Induction over the substitution.

ASSUME: $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$.

PROVE: $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R} \vdash (\sigma, \mathrm{id}){:}(\mathcal{C}, \mathcal{C}'; \mathcal{L}, \mathcal{L}'; \Phi, \Phi'; \mathcal{R}_1, \mathcal{R}')$.
⟨1⟩1. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1 \vdash (\mathrm{id}){:}(\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1)$.
PROOF: By induction on each of $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1$.

⟨1⟩2. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R} \vdash (\sigma, \mathrm{id}){:}(\mathcal{C}, \mathcal{C}'; \mathcal{L}, \mathcal{L}'; \Phi, \Phi'; \mathcal{R}_1, \mathcal{R}')$
PROOF: By induction on σ with base case as above.

2.4 Usable Substitution Lemma

If $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$ and $\mathcal{C}, \mathcal{C}'; \mathcal{L}, \mathcal{L}'; \Phi, \Phi'; \mathcal{R}_1, \mathcal{R}' \vdash J$ then $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R} \vdash \sigma(J)$.

PROOF SKETCH: Apply identity extension then substitution lemma.

ASSUME: 1. $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma){:}(\mathcal{C}'; \mathcal{L}'; \Phi'; \mathcal{R}')$.
2. $\mathcal{C}, \mathcal{C}'; \mathcal{L}, \mathcal{L}'; \Phi, \Phi'; \mathcal{R}_1, \mathcal{R}' \vdash J$.

PROVE: $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R} \vdash \sigma(J)$.

$\langle 1 \rangle 1.$ $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash (\sigma, \mathrm{id}) {:} (\mathcal{C}, \mathcal{C}'; \mathcal{L}, \mathcal{L}'; \Phi, \Phi'; \mathcal{R}_1, \mathcal{R}')$.
PROOF: Apply identity extension to 1.

$\langle 1 \rangle 2.$ $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R} \vdash (\sigma, \mathrm{id})(J)$.
PROOF: Apply substitution lemma to $\langle 1 \rangle 1$.

$\langle 1 \rangle 3.$ $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R}_1, \mathcal{R} \vdash \sigma(J)$.
PROOF: $\mathrm{id}(J) = J$.

3 Progress

If $\cdot; \cdot; \cdot; \mathcal{R} \vdash e \Leftrightarrow t$ then either $\mathrm{value}(e)$ or $\forall h : R.\ \exists e', h'.\ \langle h; e \rangle \longrightarrow \langle h'; e' \rangle$.

PROOF SKETCH: Induction over the typing rules.

ASSUME: $\cdot; \cdot; \cdot; \mathcal{R} \vdash e \Leftrightarrow t$.
PROVE: either $\mathrm{value}(e)$ or $\forall h : R.\ \exists e', h'.\ \langle h; e \rangle \longrightarrow \langle h'; e' \rangle$.

4 Framing

If $\langle h_1; e \rangle \longrightarrow \langle h_1'; e' \rangle$ and h_1, h_2 disjoint then $\langle h_1 + h_2; e \rangle \longrightarrow \langle h_1' + h_2; e' \rangle$.

PROOF SKETCH: Induction over the operational rules.

ASSUME: 1. $\langle h_1; e \rangle \longrightarrow \langle h_1'; e' \rangle$.
2. h_1, h_2 disjoint.

PROVE: $\langle h_1 + h_2; e \rangle \longrightarrow \langle h_1' + h_2; e' \rangle$.

5 Type Preservation

5.1 Ty_Spine_* and Decons_Arg_* construct same substitution and return type

If $\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash\ \overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret}$ and $\overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma'; \mathit{ret}'$ then $\sigma = \sigma'$ and $\mathit{ret} = \mathit{ret}'$.
PROOF SKETCH: Induction over arg.

5.2 Type Preservation Statement and Proof

If $\cdot; \cdot; \cdot; \mathcal{R} \vdash e \Leftrightarrow t$ then $\forall h : \mathcal{R}, e', h' : \mathcal{R}'.\ \langle h; e \rangle \longrightarrow \langle h'; e' \rangle \implies \cdot; \cdot; \cdot; \mathcal{R}' \vdash e' \Leftrightarrow t$.

PROOF SKETCH: Induction over the typing rules.

ASSUME: 1. $\cdot; \cdot; \cdot; \mathcal{R} \vdash e \Leftrightarrow t$
2. arbitrary $h : \mathcal{R}, e', h' : \mathcal{R}'$
3. $\langle h; e \rangle \longrightarrow \langle h'; e' \rangle$.

PROVE: $\cdot; \cdot; \cdot; \mathcal{R}' \vdash e' \Leftrightarrow t$.

⟨1⟩1. CASE: TY_ACTION_CREATE.
LET: $pt = mem\_ptr \overset{\times}{\mapsto}_\tau pval$.
$ret = \Sigma\, y_p{:}\mathtt{loc}.\mathtt{representable}\,(\tau*, y_p) \wedge \mathtt{alignedI}\,(mem\_int, y_p) \wedge \exists\, y{:}\beta_\tau . y_p \overset{\times}{\mapsto}_\tau y \otimes \mathtt{I}$.
ASSUME: 1. $\cdot;\cdot;\cdot;\cdot \vdash \mathtt{create}\,(mem\_int, \tau) \Rightarrow ret$.
2. $\langle \cdot; \mathtt{create}\,(mem\_int, \tau)\rangle \longrightarrow \langle \cdot + \{pt\}; \mathtt{done}\, mem\_ptr, pval, pt\rangle$.
PROVE: $\cdot;\cdot;\cdot;\cdot, \_{:}pt \vdash \mathtt{done}\, mem\_ptr, pval, pt \Leftarrow ret$

⟨2⟩1. $\cdot;\cdot;\cdot \vdash mem\_ptr \Rightarrow \mathtt{loc}$ by TY_PVAL_OBJ_INT and TY_PVAL_OBJ.

⟨2⟩2. $\mathtt{smt}\,(\cdot \Rightarrow \mathtt{representable}\,(\tau*, mem\_ptr) \wedge \mathtt{alignedI}\,(mem\_int, mem\_ptr))$ by construction of mem_ptr.

⟨2⟩3. $\cdot;\cdot;\cdot \vdash pval \Rightarrow \beta_\tau$ by construction of $pval$.

⟨2⟩4. $\cdot;\cdot;\cdot;\cdot, \_{:}pt \vdash pt \Leftarrow pt$ by TY_RES_POINTSTO.

⟨2⟩5. By TY_TVAL_I and then ⟨2⟩4 – ⟨2⟩1 with TY_TVAL_{RES,LOG,PHI,COMP} respectively, we are done.

⟨1⟩2. CASE: TY_PE_CALL.
ASSUME: 1. $\cdot;\cdot;\cdot \vdash name(\,\overline{pval_i}^{\,i}\,) \Rightarrow y{:}\beta.\ \sigma(term)$.
2. $\langle name(\,\overline{pval_i}^{\,i}\,)\rangle \longrightarrow \langle \sigma'(tpexpr){:}(y{:}\beta'.\ \sigma'(term'))\rangle$.
PROVE: $\cdot;\cdot;\cdot \vdash \sigma(tpexpr) \Leftarrow y{:}\beta.\ \sigma(term)$

⟨2⟩1. $name{:}pure\_arg \equiv \overline{x_i}^{\,i} \mapsto tpexpr \in \mathtt{Globals}$ by inversion (on either assumption).

⟨2⟩2. $\cdot;\cdot;\cdot;\cdot \vdash \overline{x_i = pval_i}^{\,i} :: pure\_arg \gg \sigma; \Sigma\, y{:}\beta.\ term \wedge \mathtt{I}$ by inversion on 1.

⟨2⟩3. $\beta = \beta'$, $term = term'$ and $\sigma = \sigma'$ by induction on $pure\_arg$.
Follows from lemma 5.1.

⟨2⟩4. $\cdot;\cdot;\cdot;\cdot \vdash (\sigma){:}(\mathcal{C};\cdot;\Phi;\cdot)$.
PROOF: Constructing such a substitution requires $\overline{\cdot;\cdot;\cdot \vdash pval_i \Rightarrow \beta_i}^{\,i}$ for each $x_i{:}\beta_i \in \mathcal{C}$ which can be deduced from ⟨2⟩2.

⟨2⟩5. $\mathcal{C}'';\cdot;\Phi'' \vdash tpexpr \Leftarrow y{:}\beta''.term''$ where $\overline{x_i}^{\,i} :: pure\_arg \rightsquigarrow \mathcal{C}'';\cdot;\Phi'';\cdot \mid \Sigma\, y{:}\beta''.term'' \wedge \mathtt{I}$ formalises the assumption that all global functions and labels are well-typed.

⟨2⟩6. $\mathcal{C} = \mathcal{C}''$, $\Phi = \Phi''$, $\beta = \beta''$ and $term = term''$.
PROOF: By induction on $pure\_arg$.

⟨2⟩7. Apply usable substitution lemma to ⟨2⟩4 and ⟨2⟩5 to finish proof.

⟨1⟩3. CASE: TY_MEMOP_PTRVALIDFORDEREF.
LET: $pt = mem\_ptr \overset{\checkmark}{\mapsto}_\tau \_$.
$ret = \Sigma\, y{:}\mathtt{bool}.\ y = \mathtt{aligned}\,(\tau, mem\_ptr) \wedge pt \otimes \mathtt{I}$.
ASSUME: 1. $\cdot;\cdot;\cdot;\mathcal{R} \vdash \mathtt{ptrValidForDeref}\,(\tau, mem\_ptr, pt) \Rightarrow ret$.
2. $\langle \cdot{+}\{pt\}; \mathtt{ptrValidForDeref}\,(\tau, mem\_ptr, pt)\rangle \longrightarrow \langle \cdot{+}\{pt\}; \mathtt{done}\, bool\_value, pt\rangle$.
PROVE: $\cdot;\cdot;\cdot;\mathcal{R} \vdash \mathtt{done}\, bool\_value, pt \Leftarrow ret$

⟨2⟩1. $\cdot;\cdot;\cdot;\mathcal{R} \vdash pt \Leftarrow pt$, by inversion on 1.

⟨2⟩2. $R = \cdot, \_{:}pt$, by TY_RES_POINTSTO.

⟨2⟩3. $bool\_value = \mathtt{aligned}\,(\tau, mem\_ptr)$ by construction of $bool\_value$.

⟨2⟩4. $\cdot;\cdot;\cdot \vdash \mathit{bool\_value} \Rightarrow \texttt{bool}$ by Ty_PVal_{True,False}.

⟨2⟩5. By Ty_TVal_I, and then ⟨2⟩2 – ⟨2⟩4 with Ty_TVal_{Res,Phi,Comp} respectively, we are done.

6 Typing Judgements

object_value_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{object\_value} \Rightarrow \texttt{obj}\,\beta$
pval_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pval} \Rightarrow \beta$
res_jtype	$::=$	
	$\|$	$\Phi \vdash \mathit{res} \;\equiv\; \mathit{res}'$
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{res\_term} \Leftarrow \mathit{res}$
spine_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \;\overline{x_i = \mathit{spine\_elem}_i}^{\,i} :: \mathit{arg} \gg \sigma; \mathit{ret}$
pexpr_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{pexpr} \Rightarrow \mathit{ident}{:}\beta.\, \mathit{term}$
tpval_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpval} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term}$
tpexpr_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi \vdash \mathit{tpexpr} \Leftarrow \mathit{ident}{:}\beta.\, \mathit{term}$
action_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_action} \Rightarrow \mathit{ret}$
memop_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{mem\_op} \Rightarrow \mathit{ret}$
seq_expr_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_expr} \Rightarrow \mathit{ret}$
is_expr_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_expr} \Rightarrow \mathit{ret}$
tval_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{tval} \Leftarrow \mathit{ret}$
texpr_jtype	$::=$	
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{seq\_texpr} \Leftarrow \mathit{ret}$
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{is\_texpr} \Leftarrow \mathit{ret}$
	$\|$	$\mathcal{C}; \mathcal{L}; \Phi; \mathcal{R} \vdash \mathit{texpr} \Leftarrow \mathit{ret}$

7 Opsem Judgements

$$
\begin{array}{lcl}
\mathit{pure\_opsem\_jtype} & ::= & \\
& | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{pexpr}' \rangle \\
& | & \langle \mathit{pexpr} \rangle \longrightarrow \langle \mathit{tpexpr}{:}(y{:}\beta.\,\mathit{term}) \rangle \\
& | & \langle \mathit{tpexpr} \rangle \longrightarrow \langle \mathit{tpexpr}' \rangle
\end{array}
$$

$$
\begin{array}{lcl}
\mathit{opsem\_jtype} & ::= & \\
& | & \langle \mathit{seq\_expr} \rangle \longrightarrow \langle \mathit{texpr}{:}\mathit{ret} \rangle \\
& | & \langle h; \mathit{seq\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
& | & \langle h; \mathit{mem\_op} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
& | & \langle h; \mathit{mem\_action} \rangle \longrightarrow \langle h'; \mathit{tval} \rangle \\
& | & \langle h; \mathit{is\_expr} \rangle \longrightarrow \langle h'; \mathit{is\_expr}' \rangle \\
& | & \langle h; \mathit{is\_texpr} \rangle \longrightarrow \langle h'; \mathit{texpr} \rangle \\
& | & \langle h; \mathit{texpr} \rangle \longrightarrow \langle h'; \mathit{texpr}' \rangle
\end{array}
$$