Robbins, Ed, King, Andy and Schrijvers, Tom (2015) ***Proof appendix to accompany the paper, "From MinX to MinC: Semantics-Driven Decompilation of Recursive Datatypes".*** **University of Kent**

A. Proof Appendix

A.1 Type Safety

We write $\Sigma; \Psi \vdash \sigma; \pi$ to signify that

$$\forall (a : \theta) \in \Psi \,.\, \Sigma; \Psi; \sigma; \pi \vdash a : \theta$$

We also write $\Gamma_c; \Sigma; \Psi \vdash \rho$ to signify that

$$\forall (x : \theta) \in \Gamma_c \,.\, \Sigma; \Psi \vdash \rho(x) : \theta * \wedge \rho(x) \neq 0$$

Moreover, we write $\Gamma_c; \Sigma \vdash \lambda_c$ to signify that

$$\forall s \in \mathit{range}(\lambda_c).\ \Gamma_c; \Sigma \vdash s$$

Proposition 8 (safety for lvalue evaluation)**.**

1. Progress: if
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$
   - $\Gamma_c; \Sigma \vdash \ell : \theta$

   then
   (a) $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \ell \rangle \xrightarrow{\ell} \langle \sigma', \pi', a \rangle$ or
   (b) $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \ell \rangle \xrightarrow{\ell} \mathsf{err}$.
2. Preservation: if
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$
   - $\Gamma_c; \Sigma \vdash \ell : \theta$
   - $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \ell \rangle \xrightarrow{\ell} \langle \sigma', \pi', a \rangle$

   then for some $\Psi' \supseteq \Psi$
   (a) $\Gamma_c; \Sigma; \Psi' \vdash \rho$
   (b) $\Sigma; \Psi' \vdash \sigma'; \pi'$
   (c) $\Sigma; \Psi' \vdash a : \theta*$

Proposition 9 (safety for expression evaluation)**.**

1. Progress: if
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$
   - $\Gamma_c; \Sigma \vdash e : \theta$

   then
   (a) $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e \rangle \xrightarrow{e} \langle \sigma', \pi', v \rangle$ or
   (b) $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e \rangle \xrightarrow{e} \mathsf{err}$.
2. Preservation: if
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$
   - $\Gamma_c; \Sigma \vdash e : \theta$
   - $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e \rangle \xrightarrow{e} \langle \sigma', \pi', v \rangle$

   then for some $\Psi' \supseteq \Psi$
   (a) $\Gamma_c; \Sigma; \Psi' \vdash \rho$
   (b) $\Sigma; \Psi' \vdash \sigma'; \pi'$
   (c) $\Sigma; \Psi' \vdash v : \theta$

Proposition 10 (safety for statement evaluation)**.**

1. Progress: if
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$
   - $\Gamma_c; \Sigma \vdash s$
   - $\Gamma_c; \Sigma \vdash \lambda_c$

   then
   (a) $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, s \rangle \xrightarrow{s} \langle \sigma', \pi', s' \rangle$ or
   (b) $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, s \rangle \xrightarrow{s} \mathsf{err}$ or
   (c) $s = \mathsf{return}$.
2. Preservation: if
   - $\Gamma_c; \Sigma \vdash s$
   - $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, s \rangle \xrightarrow{s} \langle \sigma', \pi', s' \rangle$
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$

   then for some $\Psi' \supseteq \Psi$
   (a) $\Gamma_c; \Sigma; \Psi' \vdash \rho$
   (b) $\Sigma; \Psi' \vdash \sigma'; \pi'$
   (c) $\Gamma_c; \Sigma \vdash s'$

Proposition 11 (safety for function definitions)**.**

1. Progress: if
   - $\Sigma \vdash f(\overrightarrow{x : \theta})\langle \overrightarrow{y : \theta'}, l, \lambda_c, j \rangle$
   - $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \lambda_c(l) \rangle \xrightarrow{s}{}^{*} \langle \sigma', \pi', \mathsf{return} \rangle$
   - $\Gamma_c = \{\overrightarrow{x : \theta}, \overrightarrow{y : \theta'}\}$
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$

   then
   (a) $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \lambda_c(l) \rangle \xrightarrow{s}{}^{*} \langle \sigma', \pi', \mathsf{return} \rangle$ or
   (b) $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \lambda_c(l) \rangle \xrightarrow{s}{}^{*} \mathsf{err}$ (we assume this subsumes divergence).
2. Preservation: if
   - $\Sigma \vdash f(\overrightarrow{x : \theta})\langle \overrightarrow{y : \theta'}, l, \lambda_c, j \rangle$
   - $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \lambda_c(l) \rangle \xrightarrow{s}{}^{*} \langle \sigma', \pi', \mathsf{return} \rangle$
   - $\Gamma_c = \{\overrightarrow{x : \theta}, \overrightarrow{y : \theta'}\}$
   - $\Gamma_c; \Sigma; \Psi \vdash \rho$
   - $\Sigma; \Psi \vdash \sigma; \pi$

   then for some $\Psi' \supseteq \Psi$
   (a) $\Gamma_c; \Sigma; \Psi' \vdash \rho$
   (b) $\Sigma; \Psi' \vdash \sigma'; \pi'$

Proof 1. Propositions 8, 9, 10 and 11 proved together by mutual structural induction on the typing judgements for ℓ, e, s and d_c.

- By case analysis on $\Gamma_c; \Sigma \vdash \ell : \theta$ in Fig. 4. To show 1b or conversely 1a, 2a, 2b and 2c hold for proposition 8. Observe that 2a holds if $\Psi' \supseteq \Psi$.

  1. Let $\ell = x$. By rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \rangle \xrightarrow{\ell} \langle \sigma, \pi, a \rangle$ where $a = \rho(x)$ hence 1a holds. Put $\Psi' = \Psi$. Since $\Gamma_c; \Sigma; \Psi \vdash \rho$ it follows $\Sigma; \Psi' \vdash \rho(x) : \theta*$ and 2c holds. Moreover $\Sigma; \Psi' \vdash \sigma; \pi$ and 2b holds.
  2. Let $\ell : \theta = *x : \tau$. Since $\Gamma_c; \Sigma; \Psi \vdash \rho$ it follows $a = \rho(x) \neq 0$. By rule l-ptr $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, *x \rangle \xrightarrow{\ell} \langle \sigma, \pi, \sigma(a) \rangle$ thus 1a holds. Put $\Psi' = \Psi$. By rule t-ptr $\Gamma_c; \Sigma \vdash x : \tau*$ and by $\Gamma_c; \Sigma; \Psi \vdash \rho$ it follows $\Sigma; \Psi \vdash a : \tau**$. By rule vt-addr $(a : \tau*) \in \Psi$ and by $\Sigma; \Psi \vdash \sigma; \pi$ it follows $\Sigma; \Psi; \sigma; \pi \vdash a : \tau*$. By rule st-comp $\Sigma; \Psi \vdash \sigma(a) : \tau*$ thus $\Sigma; \Psi' \vdash \sigma(a) : \tau*$ and 2c holds. Moreover $\Sigma; \Psi' \vdash \sigma; \pi$ and 2b holds.
  3. Let $\ell : \theta = x \to c : \theta_c$. Since $\Gamma_c; \Sigma; \Psi \vdash \rho$ let $a = \rho(x) \neq 0$ and let $v = \sigma(a) +_{\perp} c$. If $\rho(x) = 0$ or $v \notin \cup\pi$ then 1b holds. Otherwise $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \to c \rangle \xrightarrow{\ell} \langle \sigma, \pi, v \rangle$ and 1a holds. Put $\Psi' = \Psi$. By rule t-fld $\Gamma_c; \Sigma \vdash x : N*$ and by rule t-var $(x : N*) \in \Gamma_c$ and by $\Gamma_c; \Sigma; \Psi \vdash \rho$ it follows $\Sigma; \Psi \vdash \rho(x) : N**$. By rule vt-addr $(\rho(x) : N*) \in \Psi$ and by $\Sigma; \Psi \vdash \sigma; \pi$ it follows $\Sigma; \Psi; \sigma; \pi \vdash \rho(x) : N*$ and by rule st-comp $\Sigma; \Psi \vdash \sigma(\rho(x)) : N*$. By rule vt-addr $(\sigma(\rho(x)) : N) \in \Psi$ and by $\Gamma_c; \Sigma; \Psi \vdash \rho$ it follows $\Sigma; \Psi; \sigma; \pi \vdash \sigma(\rho(x)) : N$ and by rule st-fld $\Sigma; \Psi \vdash \sigma(\sigma(\rho(x)) + c) : \theta_c$. By rule st-comp $\Sigma; \Psi; \sigma; \pi \vdash \sigma(\rho(x)) + c : \theta_c$ and by $\Gamma_c; \Sigma; \Psi \vdash \rho$ it follows $(\sigma(\rho(x)) + c : \theta_c) \in \Psi$ and by rule vt-addr

$$\boxed{\Sigma \vdash \theta} \qquad \frac{}{\Sigma \vdash \mathsf{short}} \qquad \frac{}{\Sigma \vdash \mathsf{long}} \qquad \frac{\Sigma \vdash \tau}{\Sigma \vdash \tau*} \qquad \frac{N \in \Sigma}{\Sigma \vdash N}$$

$$\boxed{\Sigma \vdash \mathsf{decls} \xrightarrow{d} \Sigma'} \qquad \frac{}{\Sigma \vdash \epsilon \xrightarrow{d} \Sigma} \qquad \frac{\begin{array}{c}\Sigma(N) = \bot \vee N \notin dom(\Sigma)\\ \Sigma' = \Sigma \circ \{N \mapsto \vec{\theta}\}\\ \forall \theta_i \in \vec{\theta}.(\Sigma' \vdash \theta_i)\\ \Sigma' \vdash \mathsf{decls} \xrightarrow{d} \Sigma''\end{array}}{\Sigma \vdash \mathsf{struct}\ N(\vec{\theta}); \mathsf{decls} \xrightarrow{d} \Sigma''} \qquad \frac{\begin{array}{c}N \notin dom(\Sigma) \qquad \Sigma' = \Sigma \circ \{N \mapsto \bot\}\\ \Sigma' \vdash \mathsf{decls} \xrightarrow{d} \Sigma''\end{array}}{\Sigma \vdash \mathsf{struct}\ N; \mathsf{decls} \xrightarrow{d} \Sigma''}$$

Figure 13: Well-formed type declarations of MINC programs

$\Sigma; \Psi \vdash \sigma(\rho(x)) + c : \theta_c*$ and 2c holds since $\Psi' = \Psi$. Moreover $\Sigma; \Psi' \vdash \sigma; \pi$ and 2b holds.

4. Let $\ell = x[e']$. By rule t-ar $\Gamma_c; \Sigma \vdash e' : t$ hence by mutual induction:
   - Either $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e' \rangle \xrightarrow{e}$ err. By rule e-lval-err $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x[e'] \rangle \xrightarrow{e}$ err. Hence 1b.
   - Or $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e' \rangle \xrightarrow{e} \langle \sigma', \pi', v \rangle$. If $\rho(x) = 0$ then 1a holds by rule e-lval-err. Otherwise let $a = \sigma'(\rho(x)) +_{\bot} v$. If $a \notin \cup\pi'$ then 1a holds. Otherwise by rule l-ar $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x[e'] \rangle \xrightarrow{\ell} \langle \sigma', \pi', a \rangle$. Hence 1a holds.
     By induction there exists $\Psi' \supseteq \Psi$ such that $\Sigma; \Psi' \vdash \sigma'; \pi'$. By rule t-ar $\Gamma_c; \Sigma \vdash x : \theta[]*$ and by rule t-var $(x : \theta[]*) \in \Gamma_c$ and by $\Gamma_c; \Sigma; \Psi' \vdash \rho$ it follows $\Sigma; \Psi' \vdash \rho(x) : \theta[]**$. By rule vt-addr $(\rho(x) : \theta[]*) \in \Psi'$ and by $\Sigma; \Psi' \vdash \sigma'; \pi'$ it follows $\Sigma; \Psi'; \sigma'; \pi' \vdash \rho(x) : \theta[]*$ and by rule st-comp $\Sigma; \Psi' \vdash \sigma'(\rho(x)) : \theta[]*$. By rule vt-addr $(\sigma'(\rho(x)) : \theta[]) \in \Psi'$ and by $\Gamma_c; \Sigma; \Psi' \vdash \rho$ it follows $\Sigma; \Psi'; \sigma'; \pi' \vdash \sigma'(\rho(x)) : \theta[]$ and by rule st-ar $\Sigma; \Psi' \vdash \sigma'(\sigma'(\rho(x)) + v) : \theta$. By rule st-comp $\Sigma; \Psi'; \sigma'; \pi' \vdash \sigma'(\rho(x)) + v : \theta$ and by $\Gamma_c; \Sigma; \Psi' \vdash \rho$ it follows $(\sigma'(\rho(x)) + v : \theta) \in \Psi'$ and by rule vt-addr $\Sigma; \Psi' \vdash \sigma'(\rho(x)) + v : \theta*$ and 2c holds. Moreover $\Sigma; \Psi' \vdash \sigma; \pi$ and 2b holds.

- By case analysis on $\Gamma_c; \Sigma \vdash e : \theta$ in Fig. 4. To show that either 1b or conversely 1a, 2a 2b and 2c of Proposition 9 hold. Observe that 2a holds if $\Psi' \supseteq \Psi$.

1. Let $e : \theta = \&x : \tau*$. By rule t-amp $\Gamma_c; \Sigma \vdash x : \tau$ thus $(x : \tau) \in \Gamma_c$ and by $\Gamma_c; \Sigma; \Psi \vdash \rho$ it follows $\Sigma; \Psi \vdash a : \tau*$ where $a = \rho(x) \neq 0$. By rule e-amp $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \&x \rangle \xrightarrow{e} \langle \sigma, \pi, a \rangle$ hence 1a holds. Put $\Psi' = \Psi$ thus $\Sigma; \Psi' \vdash a : \tau*$ and 2c holds whilst 2b is immediate.
2. Let $e : \theta = c_l : \mathsf{long}$. By rule e-const $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, c_l \rangle \xrightarrow{e} \langle \sigma, \pi, c_l \rangle$. Hence 1a.
   Let $\Psi' = \Psi$. By rule vt-l $\Sigma; \Psi \vdash c_l : \mathsf{long}$. Hence 2c. Also 2b.
3. Let $e : \theta = c_s : \mathsf{short}$. By rule e-const $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, c_s \rangle \xrightarrow{e} \langle \sigma, \pi, c_s \rangle$. Hence 1a.
   Let $\Psi' = \Psi$. By rule vt-s $\Sigma; \Psi \vdash c_s : \mathsf{short}$. Hence 2c. Also 2b.
4. Let $e : \theta = 0_l : \tau*$. By rule e-const $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, 0_l \rangle \xrightarrow{e} \langle \sigma, \pi, 0_l \rangle$. Hence 1a.
   Let $\Psi' = \Psi$. By rule vt-null $\Sigma; \Psi \vdash c_s : \tau*$. Hence 2c. Also 2b.
5. Let $e : \theta = \mathsf{new}\ \tau : \tau*$. By rule e-new $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{new}\ \tau \rangle \xrightarrow{e} \langle \sigma', \pi, a \rangle$ where $\sigma' = \sigma \circ \{a \mapsto \bot\}$. Hence 1a.
   Let $\Psi' = \Psi \circ \{a \mapsto \tau\}$. By rule vt-addr $\Sigma; \Psi \vdash a : \tau*$ hence 2c. Also by rule vt-bot $\Sigma; \Psi' \vdash \bot : \tau$ by and rule st-comp $\Sigma; \Psi'; \sigma'; \pi \vdash a : \tau$ hence $\Sigma; \Psi' \vdash \sigma'; \pi$ and 2b holds.
6. Let $e : \theta = \mathsf{new\ struct}\ N : N*$ and $n = |\Sigma(N)|$. By rule e-str $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{new\ struct}\ N \rangle \xrightarrow{e} \langle \sigma', \pi', a \rangle$ where $\sigma' = \sigma \circ \{a \mapsto \bot, \ldots, a + n - 1 \mapsto \bot\}$ and $\pi' = \pi \cup \{[a, a + n - 1]\}$. Put $\Psi' = \Psi \cup \{a : N, a + 1 : \theta_1, \ldots, a + n - 1 : \theta_{n-1}\}$. By rule vt-addr $\Sigma; \Psi' \vdash a : N*$ hence 2c holds.
   Let $i \in [0, n-1]$. Then $\sigma'(a+i) = \bot$ hence $\Sigma; \Psi' \vdash \sigma'(a + i) : \theta_i$ by rule vt-bot therefore $\Sigma; \Psi'; \sigma'; \pi' \vdash a + i : \theta_i$. By rule st-fld $\Sigma; \Psi'; \sigma'; \pi' \vdash a : N$ hence 2b holds.
7. Let $e : \theta = \mathsf{new}\ \theta[e] : \theta[]*$. By rule t-new-ar $\Gamma_c; \Sigma \vdash e : t$ hence by induction:
   - Either $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e \rangle \xrightarrow{e}$ err. By rule e-ar-err $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{new}\ \theta[e] \rangle \xrightarrow{e}$ err. Hence 1b.
   - Or $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e \rangle \xrightarrow{e} \langle \sigma', \pi', v \rangle$. By rule e-ar $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{new}\ \theta[e] \rangle \xrightarrow{e} \langle \sigma'', \pi'', a \rangle$ where $\sigma'' = \sigma' \circ \{a \mapsto \bot, \ldots, a + v - 1 \mapsto \bot\}$. Hence 1a.
     By induction there exists $\Phi' \supseteq \Phi$ such that $\Sigma; \Psi' \vdash \sigma'; \pi'$. Put $\Psi'' = \Psi' \circ \{a \mapsto \theta[], \ldots, a + v - 1 \mapsto \theta[]\}$. By rule vt-addr it follows $\Sigma; \Psi'' \vdash a : \theta[]*$ hence 2c. By rule vt-bot it follows $\Sigma; \Psi'' \vdash \bot : \theta[]$ and by st-comp it follows $\Sigma; \Psi''; \sigma''; \pi'' \vdash a + i : \theta[]$ for all $i \in [0, v - 1]$ hence 2b.
8. Let $e : \theta = (e_1 \oplus e_2) : t$. By rule t-$\otimes$ $\Gamma_c; \Sigma \vdash e_1 : t$ and $\Gamma_c; \Sigma \vdash e_2 : t$. Hence by induction:
   - Either $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e_1 \rangle \xrightarrow{e}$ err. By rule e-op-err$_1$ $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (e_1 \oplus e_2) \rangle \xrightarrow{e}$ err. Hence 1b.
   - Or $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma', \pi', e_2 \rangle \xrightarrow{e}$ err. Like previous case.
   - Or $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e_1 \rangle \xrightarrow{e} \langle \sigma', \pi', v_1 \rangle$ and $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma', \pi', e_2 \rangle \xrightarrow{e} \langle \sigma'', \pi'', v_2 \rangle$.
     - Either $v_1 \oplus_\pi v_2 =$ err. By rule e-op-err$_3$ $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (e_1 \oplus e_2) \rangle \xrightarrow{e}$ err. Hence 1b.
     - Or $v_1 \oplus_\pi v_2 = v$. By rule e-op $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (e_1 \oplus e_2) \rangle \xrightarrow{e} \langle \sigma', \pi, v \rangle$. Hence 1a.
       By induction $\Sigma; \Psi'' \vdash v_1 : t$ and $\Sigma; \Psi'' \vdash v_2 : t$. If $t = \mathsf{short}$ then $v = \bot$ or $v = n_s$ where $n \in [-2^{15}, 2^{15} - 1]$. If $v = \bot$ then $\Sigma; \Psi'' \vdash v : \mathsf{short}$. by rule vt-bot. Otherwise if $v = n_s$ then $\Sigma; \Psi'' \vdash v : \mathsf{short}$ by rule vt-s. An analgous argument holds if $t = \mathsf{long}$ hence 2c. Also 2b trivially by induction.
9. Let $e : \theta = (e_1 \oplus e_2) : \tau[]*$. Similar to previous case.

10. Let $e : \theta = f(\vec{e}) : \theta_j$. By rule t-call $\Gamma_c; \Sigma \vdash e_i : \theta'_i$ where $\phi_c(f) = f(\overrightarrow{x : \theta})\langle \overrightarrow{y : \theta''}, l, \lambda_c, j\rangle$ and $\Sigma \vdash \vec{\theta'} <: \vec{\theta}$. With respect to e_i there are two possibilities:
    - Either for some i: $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma_{i-1}, \pi_{i-1}, e_i\rangle \xrightarrow{e} \mathsf{err}$. Then by rule e-call-err it follows that 1b holds.
    - Or for all i: $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma_{i-1}, \pi_{i-1}, e_i\rangle \xrightarrow{e} \langle \sigma_i, \pi_i, v_i\rangle$ and by the inductive hypothesis $\Sigma; \Psi_i \vdash \theta_i : v_i$ and $\Sigma; \Psi_i \vdash \sigma_i; \pi_i$. Let $\Psi' = \Psi_n \cup \{\overrightarrow{a : \theta}, \overrightarrow{a' : \theta'}\}$. Then it is easy to verify $\Sigma; \Psi' \vdash \sigma'; \pi_n$ and $\Gamma_c; \Sigma; \Psi' \vdash \rho'$. By the progress induction hypothesis we then have for s:
      - Either $\Sigma; \lambda_c; \vec{\rho}; \rho; \rho' \vdash \langle \sigma', \pi_n, \lambda_c(l)\rangle \xrightarrow{s}{}^* \langle \sigma'', \pi', \mathsf{return}\rangle$. Hence $1a$.
      - Otherwise 1b.

      Preservation follows from the induction hyptheses for all e_i and s.

- By case analysis on $\Gamma_c; \Sigma \vdash s$ in Fig. 4. To show that either 1b or conversely 1a, 2a, 2b and 2c of Proposition 10 hold. Observe that 2a holds if $\Psi' \supseteq \Psi$.
  1. Let $\Gamma_c; \Sigma \vdash (\ell := e); s$. From the induction hypothesis for ℓ, either $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \ell\rangle \xrightarrow{\ell} \mathsf{err}$, and hence 1b, or $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \ell\rangle \xrightarrow{\ell} \langle \sigma', \pi', a\rangle$. In the latter case, we have either $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma', \pi', e\rangle \xrightarrow{e} \mathsf{err}$, and hence 1b, or $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma', \pi', e\rangle \xrightarrow{e} \langle \sigma'', \pi'', v\rangle$. By s-assn we then have $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (\ell := e); s\rangle \xrightarrow{s} \langle \sigma''', \pi'', s\rangle$ where $\sigma''' = \sigma'' \circ \{a \mapsto v\}$ and hence 1a.
     We get $\Gamma_c; \Sigma \vdash s$ from t-assn. Hence 2c. From the induction hypotheses for ℓ and e we get type preservations $\Sigma; \Psi'' \vdash a : \theta_1*$ and $\Sigma; \Psi'' \vdash v : \theta_2$ and type consistency $\Sigma; \Psi'' \vdash \sigma''; \pi''$. Hence, through rule vt-addr we know that $(a : \theta_1) \in \Psi''$. From rule t-assn we know $\Sigma \vdash \theta_2 <: \theta_1$. Hence, through rule vt-subt we have $\Sigma; \Psi'' \vdash v : \theta_1$. Since $\sigma'''(a) = v$ we have hence by rule st-comp $\Sigma; \Psi''; \sigma'''; \pi'' \vdash a : \theta_1$. Hence $\Sigma; \Psi'' \vdash \sigma'''; \pi''$. Thus 2b.
  2. Let $\Gamma_c; \Sigma \vdash (\mathsf{if}\ e\ \mathsf{goto}\ l); s$. Then
     - Either $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e\rangle \xrightarrow{e} \mathsf{err}$. Hence 1b.
     - Or $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, e\rangle \xrightarrow{e} \langle \sigma', \pi', v\rangle$. Then
       - Either $v = \bot$. Hence 1b.
       - Or $v = 0$. Then by rule s-if-false $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (\mathsf{if}\ e\ \mathsf{goto}\ l); s\rangle \xrightarrow{s} \langle \sigma', \pi s,' \rangle$. Hence 1a. We call this scenario 1.
       - Or $v \neq 0 \wedge v \neq \bot$. Then
         - Either $l \notin dom(\lambda_c)$. Then 1b.
         - Or $s' = \lambda_c(l)$. Then by rule s-if-true $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (\mathsf{if}\ e\ \mathsf{goto}\ l); s\rangle \xrightarrow{s} \langle \sigma', \pi s','\rangle$. Hence 1a. We call this scenario 2.

     In scenario 1 we have from t-if $\Gamma_c; \Sigma \vdash s$. Hence 2c. In scenario 2 we have that $s' \in range(\lambda_c)$. Hence $\Gamma_c; \Sigma \vdash s'$. Hence 2c. In both scenarios we have from the induction hypthesis for e that $\Sigma; \Psi' \vdash \sigma'; \pi'$. Hence 2b.
  3. Let $\Gamma_c; \Sigma \vdash \mathsf{goto}\ l$. Then either $l \notin dom(\lambda_c)$ and thus $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{goto}\ l\rangle \xrightarrow{s} \mathsf{err}$. Hence 1b. Alternatively $\lambda_c(l) = s$. Then by rule s-goto $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{goto}\ l\rangle \xrightarrow{s} \langle \sigma, \pi, s\rangle$. Hence 1a.
     From $\Gamma_c; \Sigma \vdash \lambda_c$ it follows that $\Gamma_c; \Sigma \vdash s$. Hence 2c. Let $\Psi' = \Psi$. Then 2b.
  4. Let $\Gamma_c; \Sigma \vdash \mathsf{return}$. Hence 1c. Also vacuously 2c and 2b.
- Proposition 11 follows by the repeated application of Proposition 10 combining progress and preservation at every step.
  Besides the givens of Proposition **??**, Proposition 10 also requires $\Gamma_c; \Sigma \vdash \lambda_c$. This is given by rule t-def which is the only possible way that the well-typing of the function definition could have been constructed.

A.2 Well-Typed Decompilation

Proposition 12 (well-typed instruction decompilation)**.** If $\mu_\Gamma; \Gamma_c; \Sigma \vdash \iota \overset{\iota}{\rightsquigarrow} \ell := e$ then for some θ_1 and θ_2

1. $\Gamma_c; \Sigma \vdash \ell : \theta_1$
2. $\Gamma_c; \Sigma \vdash e : \theta_2$
3. $\Sigma \vdash \theta_2 <: \theta_1$

Proof 2. The proof proceeds by case analysis on the inference rules of the instruction translation relation.

1. Case tr-$\oplus$-r$*_1$. Let $\theta_1 = \theta_2 = \theta[]*$. From tr-$\oplus$-r$*_1$ we have $(x : \theta[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta[]*$. Hence 1. From tr-$\oplus$-r$*_1$ we have $\Gamma_c; \Sigma \vdash m : \mathsf{long}$. From tr-$\oplus$-r$*_1$ we have $(y : \mathsf{long}) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \mathsf{long}$. From both of these we get by rule t-$\otimes$ $\Gamma_c; \Sigma \vdash y * m : \mathsf{long}$. From that and the type of x we get through rule t-ptr-$\oplus$ $\Gamma_c; \Sigma \vdash x \oplus (y * m) : \theta[]*$. Hence 2. From rule sub-refl 3.
2. Case tr-$\oplus$-r$*_2$. Let $\theta_1 = \theta_2 = t$. From tr-$\oplus$-r$*_2$ we have $(x : t) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : t$. Hence 1. From tr-$\oplus$-r$*_2$ we have $\Gamma_c; \Sigma \vdash c : t$. From tr-$\oplus$-r$*_1$ we have $(y : t) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : t$. From both of these we get by rule t-$\otimes$ $\Gamma_c; \Sigma \vdash y * c : t$. From that and the type of x we get through rule t-$\otimes$ $\Gamma_c; \Sigma \vdash x \oplus (y * c) : t$. Hence 2. From rule sub-refl 3.
3. Case tr-$\otimes$-rc. Let $\theta_1 = \theta_2 = t$. From tr-$\otimes$-rc we have $(x : t) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : t$. Hence 1. From tr-$\otimes$-rc we have $\Gamma_c; \Sigma \vdash c : t$. From that and the previous $\Gamma_c; \Sigma \vdash x : t$ we have by rule t-$\otimes$ $\Gamma_c; \Sigma \vdash x \otimes c : t$. Hence 2. From rule sub-refl 3.
4. Case tr-$\otimes$-rr. Let $\theta_1 = \theta_2 = t$. From tr-$\otimes$-rr we have $(x : t) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : t$. Hence 1. From tr-$\otimes$-rr we have $(y : t) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : t$. From that and the previous $\Gamma_c; \Sigma \vdash x : t$ we have by rule t-$\otimes$ $\Gamma_c; \Sigma \vdash x \otimes y : t$. Hence 2. From rule sub-refl 3.
5. Case tr-$\oplus$-rc. Let $\theta_1 = \theta_2 = \theta[]*$. From tr-$\oplus$-rc we have $(x : \theta[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta[]*$. Hence 1. From tr-$\oplus$-rc we have $\Gamma_c; \Sigma \vdash m : t$. From that and the previous $\Gamma_c; \Sigma \vdash x : \theta[]*$ we have by rule t-ptr-$\oplus$ $\Gamma_c; \Sigma \vdash x \oplus m : \theta[]*$. Hence 2. From rule sub-refl 3.
6. Case tr-mov-rc. Let $\theta_1 = \theta_2 = t$. From tr-mov-rc we have $(x : t) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : t$. Hence 1. From tr-mov-rc we have $\Gamma_c; \Sigma \vdash c : t$. Hence 2. From rule sub-refl 3.
7. Case tr-mov-r0. Let $\theta_1 = \theta_2 = \tau*$. From tr-mov-r0 we have $(x : \tau*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \tau*$. Hence 1. From t-null we have $\Gamma_c; \Sigma \vdash 0 : \tau*$. Hence 2. From rule sub-refl 3.
8. Case tr-mov-rr. From tr-mov-rr we have $(x : \theta_1) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_1$. Hence 1. From tr-mov-rr we have $(y : \theta_2) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2$. Hence 2. From tr-mov-rr we have $\Sigma \vdash \theta_2 <: \theta_1$. Hence 3.
9. Case tr-mov-ri$_1$. From tr-mov-ri$_1$ we have $(x : \theta_1) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_1$. Hence 1. From tr-mov-ri$_1$ we have $(y : \theta_2*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2*$. Then by rule t-ptr $\Gamma_c; \Sigma \vdash *y : \theta_2$. Hence 2. From tr-mov-ri$_1$ we have $\Sigma \vdash \theta_2 <: \theta_1$. Hence 3.
10. Case tr-mov-ir$_1$. From tr-mov-ir$_1$ we have $(x : \theta_1*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_1*$. Then by rule t-ptr $\Gamma_c; \Sigma \vdash *x : \theta_1$. Hence 1. From tr-mov-ir$_1$ we have $(y : \theta_2) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2$. Hence 1. From tr-mov-ir$_1$ we have $\Sigma \vdash \theta_2 <: \theta_1$. Hence 3.

11. Case tr-mov-ri$_2$. From tr-mov-ri$_2$ we have $(x : \theta_1) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_1$. Hence 1. From tr-mov-ri$_2$ we have $(y : \theta_2[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2[]*$. Also by rule t-l$\Gamma_c; \Sigma \vdash 0 : \mathsf{long}$. Then by rule t-ar$\Gamma_c; \Sigma \vdash y[0] : \theta_2$. Hence 2. From tr-mov-ri$_2$ we have $\Sigma \vdash \theta_2 <: \theta_1$. Hence 3.
12. Case tr-mov-ir$_2$. From tr-mov-ir$_2$ we have $(x : \theta_1[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_1[]*$. Also by rule t-l$\Gamma_c; \Sigma \vdash 0 : \mathsf{long}$. Then by rule t-ar $\Gamma_c; \Sigma \vdash x[0] : \theta_1$. Hence 1. From tr-mov-ir$_2$ we have $(y : \theta_2) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2$. Hence 2. From tr-mov-ir$_2$ we have $\Sigma \vdash \theta_2 <: \theta_1$. Hence 3.
13. Case tr-mov-ri$_3$. From tr-mov-ri$_3$ we have $(x : \theta) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta$. Hence 1. From tr-mov-ri$_3$ we have $(y : N*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : N*$. Then by rule t-fld$\Gamma_c; \Sigma \vdash y \to 0 : \theta_0$. Hence 2. From tr-mov-ri$_3$ we have $\Sigma \vdash \theta_0 <: \theta$. Hence 3.
14. Case tr-mov-ir$_3$. From tr-mov-ir$_3$ we have $(x : N*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : N*$. Then by rule t-fld$\Gamma_c; \Sigma \vdash x \to 0 : \theta_0$. Hence 1. From tr-mov-ir$_3$ we have $(y : \theta) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta$. Hence 2. From tr-mov-ir$_3$ we have $\Sigma \vdash \theta <: \theta_0$. Hence 3.
15. Case tr-mov-ri+$_1$. From tr-mov-ri+$_1$ we have $(x : \theta_1) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_1$. Hence 1. From tr-mov-ri+$_1$ we have $(y : \theta_2[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2[]*$. Also from tr-mov-ri+$_1$ we have $\Gamma_c; \Sigma \vdash m : t$. Then by rule t-ar$\Gamma_c; \Sigma \vdash y[m] : \theta_2$. Hence 2. From tr-mov-ri+$_1$ we have $\Sigma \vdash \theta_2 <: \theta_1$. Hence 3.
16. Case tr-mov-i+r$_1$. From tr-mov-i+r$_1$ we have $(x : \theta_1[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_1[]*$. Also from tr-mov-i+r$_1$ we have $\Gamma_c; \Sigma \vdash m : t$. Then by rule t-ar$\Gamma_c; \Sigma \vdash x[m] : \theta_1$. Hence 1. From tr-mov-i+r$_1$ we have $(y : \theta_2) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2$. Hence 2. From tr-mov-i+r$_1$ we have $\Sigma \vdash \theta_2 <: \theta_1$. Hence 3.
17. Case tr-mov-ri+$_2$. From tr-mov-ri+$_2$ we have $(x : \theta) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta$. Hence 1. From tr-mov-ri+$_2$ we have $(y : N*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : N*$. Then by rule t-fld$\Gamma_c; \Sigma \vdash y \to m : \theta_m$. Hence 2. From tr-mov-ri+$_2$ we have $\Sigma \vdash \theta_m <: \theta$. Hence 3.
18. Case tr-mov-i+r$_2$. From tr-mov-i+r$_2$ we have $(x : N*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : N*$. Then by rule t-fld$\Gamma_c; \Sigma \vdash x \to m : \theta_m$. Hence 1. From tr-mov-i+r$_2$ we have $(y : \theta_2) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : \theta_2$. Hence 2. From tr-mov-i+r$_2$ we have $\Sigma \vdash \theta <: \theta_m$. Hence 3.
19. Case tr-alloc-r*. From tr-alloc-r* we have $(x : \theta[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta[]*$. Hence 1. From tr-alloc-r* we have $\Gamma_c; \Sigma \vdash m : t$. From tr-alloc-r* we have $(y : t) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash y : t$. From both of these we get by rule t-$\otimes$ $\Gamma_c; \Sigma \vdash y * m : t$. Then from t-new-ar we get $\Gamma_c; \Sigma \vdash \mathsf{new}\ \theta[y * m] : \theta[]*$. Hence 2. From rule sub-refl 3.
20. Case tr-alloc-rc$_1$. From tr-alloc-rc$_1$ we have $(x : \theta*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta*$. Hence 1. From t-new we get $\Gamma_c; \Sigma \vdash \mathsf{new}\ \theta : \theta*$. Hence 2. From rule sub-refl 3.
21. Case tr-alloc-rc$_2$. From tr-alloc-rc$_2$ we have $(x : N*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : N*$. Hence 1. From t-new-str we get $\Gamma_c; \Sigma \vdash \mathsf{new}\ N : N*$. Hence 2. From rule sub-refl 3.
22. Case tr-alloc-rc$_3$. From tr-alloc-rc$_3$ we have $(x : \theta[]*) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta[]*$. Hence 1. From tr-alloc-rc$_3$ we have $\Gamma_c; \Sigma \vdash m : t$. Then from rule t-new-ar we have $\Gamma_c; \Sigma \vdash \mathsf{new}\ \theta[m] : \theta[]*$. Hence 2. From rule sub-refl 3.
23. Case tr-call. From tr-call we have $(u : \theta_u) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash u : \theta_u$. Hence 1. We have:
    - From tr-call we have $\phi_c(f) = f(\overrightarrow{x : \theta})\langle\overrightarrow{y : \theta'}, l, \lambda_c, j\rangle$.
    - From tr-call we have $\overrightarrow{(v : \theta_v)} \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash \vec{v} : \vec{\theta}_v$.
    - From tr-call we have $\Sigma \vdash \vec{\theta}_v <: \vec{\theta}$.
    - By rule sub-reflwe have $\Sigma \vdash \theta'_j <: \theta'_j$.

    Hence by rule t-call we have $\Gamma_c; \Sigma \vdash : \theta'_j$. Hence 2. From tr-call we have $\Sigma \vdash \theta'_j <: \theta_u$. Hence 3.

Proposition 13 (well-typed block decompilation). If $\mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash b \overset{b}{\rightsquigarrow} s$ then $\Gamma_c; \Sigma \vdash s$.

Proof 3. This proof proceeds by structural induction on the block translation relation.

1. Case tr-instr. From tr-instrwe have $\mu_\Gamma; \Gamma_c; \Sigma \vdash \iota \overset{\iota}{\rightsquigarrow} \ell := e$. Hence, by Proposition 12 we have$\Gamma_c; \Sigma \vdash \ell : \theta_1$, $\Gamma_c; \Sigma \vdash e : \theta_2$ and $\Sigma \vdash \theta_2 <: \theta_1$. Also by rule tr-instr we have $\mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash b \overset{b}{\rightsquigarrow} s$. Hence by the induction hypothesis we have $\Gamma_c; \Sigma \vdash s$. Then by rule t-assn we have $\Gamma_c; \Sigma \vdash \ell := e; s$
2. Case tr-if. From tr-if we have $(x : \theta) \in \Gamma_c$. Then by rule t-var $\Gamma_c; \Sigma \vdash x : \theta_u$. Also from tr-if we have $\mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash b \overset{b}{\rightsquigarrow} s$. Hence, from the induction hypothesis we have $\Gamma_c; \Sigma \vdash s$ Then the proposition follows from rule t-if.
3. Case tr-goto. This follows from rule t-goto.
4. Case tr-ret. This follows from rule t-ret.

Proposition 14 (well-typed definition decompilation). If $\Sigma \vdash d_x \rightsquigarrow d_c$ then $\Sigma \vdash d_c$.

Proof 4. We show that the four preconditions to rule t-def are satisfied:

1. From rule tr-def we know that $\Gamma_c = \{\overrightarrow{x : \theta}, \overrightarrow{y : \theta'}\}$.
2. From rule tr-def we know that $a \in dom(\lambda_c)$ and $l = \mu_\lambda(a)$. Hence $l \in range(\mu_\lambda)$. From the rule we also know that $range(\mu_\lambda) = dom(\lambda_c)$. Hence $l \in dom(\lambda_c)$.
3. From rule tr-def we know that $r_{y_j} \in \overrightarrow{r_y}$. We also know that $y_j = \mu_\Gamma(r_{y_j})$ and that $\overrightarrow{y} = \mu_\Gamma(\overrightarrow{r_y})$. Hence $y_j \in \overrightarrow{y}$.
4. From rule tr-def we know that $\forall (a \mapsto l) \in \mu_\lambda : \mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash \lambda_x(a) \overset{b}{\rightsquigarrow} \lambda_c(l)$. From Proposition 13 we then know that $\forall l \in range(\mu_\lambda) : \Gamma_c; \Sigma \vdash \lambda_c(l)$. From rule tr-def we know that $range(\mu_\lambda) = dom(\lambda_c)$. Hence $\forall l \in dom(\lambda_c) : \Gamma_c; \Sigma \vdash \lambda_c(l)$.

Hence by rule t-def we conclude $\Sigma \vdash f(\overrightarrow{x : \theta})\langle\overrightarrow{y : \theta'}, l, \lambda_c, j\rangle$.

A.3 Semantics Preservation

Instructions We prove Propositions 7 and 6 together.

Proof 5. The proof proceeds by case analysis on the derivation of the judgement $\mu_\Gamma; \Gamma_c; \Sigma \vdash \iota \overset{\iota}{\rightsquigarrow} \ell := e$.

1. Case tr-$\oplus$-r*$_1$. Then $\iota = (\mathsf{op}_4^\oplus\ r_i, r_j * c)$, $\ell = x$ and $e = x \oplus (y * m)$.
   (a) This case is not possible. Rule ex-$\oplus$-r* always applies.
   (b) In this case rules ex-$\oplus$-r* is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{op}_4^\oplus\ r_i, r_j * c\rangle \overset{\iota}{\to} \langle H, R'\rangle$. Here $R' = R \circ_4 \{r_i \mapsto \vec{b}_i \oplus_4 (\vec{b}_j *_4 c)\}$ where $\vec{b}_i = R_{0:4}(r_i)$ and $\vec{b}_j = R_{0:4}(r_j)$. Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \overset{\ell}{\to} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-op, e-lval, l-var and e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (x \oplus (y * m))\rangle \overset{e}{\to} \langle \sigma, \pi, v\rangle$ where $v = v_x \oplus_\pi (v_y *_\pi m)$, $v_x = \sigma(a)$, $a' = \rho(y)$ and $v_y = \sigma(a')$.
   From rule tr-$\oplus$-r*$_1$ we know $(r_i : x)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_i \leftrightsquigarrow v_x$. Similarly, we know $\mu_a \vdash \vec{b}_j \leftrightsquigarrow v_y$. Then from $(x : \theta[]*) \in$

Γ_c and the store typing of σ it follows that $v_x = n_*$ and from the success of the addition, it also follows that $[n_*, n_* \oplus (v_y * m)] \subseteq\in \pi$. Hence, also from the store typing all m values at the addresses in this range have type θ. From the related heaps it then follows with $c/m = \mathit{sizeof}(\theta)$ that $\mu_a \vdash (\vec{b}_i \oplus_4 (\vec{b}_j *_4 c)) \leftrightsquigarrow (v \oplus_\pi (v_y * m))$. Hence, the update registers are still related.

2. Case tr-$\oplus$-r*$_2$. Then $\iota = (\mathsf{op}^{\oplus}_w\ r_i, r_j * c)$, $\ell = x$ and $e = x \oplus (y * c)$.
   (a) This case is not possible. Rule ex-$\oplus$-r* always applies.
   (b) In this case rules ex-$\oplus$-r* is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{op}^{\oplus}_w\ r_i, r_j * c\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}_i \oplus_w (\vec{b}_j *_w c)\}$ where $\vec{b}_i = R_{0:w}(r_i)$ and $\vec{b}_j = R_{0:w}(r_j)$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-op, e-lval, l-var and e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (x \oplus (y * m))\rangle \xrightarrow{e} \langle \sigma, \pi, v\rangle$ where $v = v_x \oplus_\pi (v_y *_\pi m)$, $v_x = \sigma(a)$, $a' = \rho(y)$ and $v_y = \sigma(a')$.
   From rule tr-$\oplus$-r*$_2$ we know $(r_i : x)_w \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_i \leftrightsquigarrow v_x$. Similarly, we know $\mu_a \vdash \vec{b}_j \leftrightsquigarrow v_y$. It then follows that $\mu_a \vdash (\vec{b}_i \oplus_w (\vec{b}_j *_w c)) \leftrightsquigarrow (v \oplus_\pi (v_y * c))$. Hence, the update registers are still related.
3. Case tr-$\otimes$-rc. Then $\iota = (\mathsf{op}^{\otimes}_w\ r_i, c)$, $\ell = x$ and $e = x \otimes c$.
   (a) This case is not possible. Rule ex-$\otimes$-rc always applies.
   (b) In this case rules ex-$\otimes$-rc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{op}^{\otimes}_w\ r_i, c\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b} \otimes_w c\}$ where $\vec{b} = R_{0:w}(r_i)$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-op, e-lval, l-var and e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (x \otimes c)\rangle \xrightarrow{e} \langle \sigma, \pi, v'\rangle$ where $v' = v \otimes_\pi c$ and $v = \sigma(a)$.
   From rule tr-$\otimes$-rc we know $(r_i : x)_w \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b} \leftrightsquigarrow v$. Then from $(x : t) \in \Gamma_c$ and $w = \mathit{sizeof}(t)$ it follows that $\mu_a \vdash (\vec{b} \otimes_w c) \leftrightsquigarrow (v \otimes_\pi c)$. Hence, the update registers are still related.
4. Case tr-$\oplus$-rc. Then $\iota = (\mathsf{op}^{\oplus}_4\ r_i, c)$, $\ell = x$ and $e = x \oplus m$.
   (a) This case is not possible. Rule ex-$\otimes$-rc always applies.
   (b) In this case rules ex-$\otimes$-rc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{op}^{\oplus}_4\ r_i, c\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_4 \{r_i \mapsto \vec{b} \oplus_4 c\}$ where $\vec{b} = R_{0:4}(r_i)$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-op, e-lval, l-var and e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (x \oplus m)\rangle \xrightarrow{e} \langle \sigma, \pi, v'\rangle$ where $v' = v \oplus_\pi m$ and $v = \sigma(a)$.
   From rule tr-$\oplus$-rc we know $(r_i : x)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b} \leftrightsquigarrow v$. Then from $(x : \theta[]*) \in \Gamma_c$ and the store typing of σ it follows that $v = n_*$ and from the success of the addition, it also follows that $[n_*, n_* \oplus m] \subseteq\in \pi$. Hence, also from the store typing all m values at the addresses in this range have type θ. From the related heaps it then follows with $c/m = \mathit{sizeof}(\theta)$ that $\mu_a \vdash (\vec{b} \oplus_4 c) \leftrightsquigarrow (v \oplus_\pi m)$. Hence, the update registers are still related.
5. Case tr-$\otimes$-rr. Then $\iota = (\mathsf{op}^{\otimes}_w\ r_i, r_j)$, $\ell = x$ and $e = x \otimes y$.
   (a) This case is not possible. Rule ex-$\otimes$-rr always applies.
   (b) In this case rules ex-$\otimes$-rc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{op}^{\otimes}_w\ r_i, r_j\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}_i \oplus_w \vec{b}_j\}$ where $\vec{b}_i = R_{0:w}(r_i)$ and $\vec{b}_j = R_{0:w}(r_j)$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-op, e-lval and l-var we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, (x \otimes y)\rangle \xrightarrow{e} \langle \sigma, \pi, v\rangle$ where $v = v_x \otimes_\pi v_y$, $v_x = \sigma(a)$, $a' = \rho(y)$ and $v_y = \sigma(a')$.
   From rule tr-$\otimes$-rr we know $(r_i : x)_w \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_i \leftrightsquigarrow v_x$. By similar reasoning we know $\mu_a \vdash \vec{b}_j \leftrightsquigarrow v_y$. Then from $(x : t) \in \Gamma_c$, $(y : t) \in \Gamma_c$ and $w = \mathit{sizeof}(t)$ it follows that $\mu_a \vdash (\vec{b}_i \otimes_w \vec{b}_j) \leftrightsquigarrow (v_x \otimes_\pi v_y)$. Hence, the update registers are still related.
6. Case tr-mov-rc. Then $\iota = (\mathsf{mov}_w\ r_i, c)$, $\ell = x$ and $e = c$.
   (a) This case is not possible. Rule ex-mov-rc always applies.
   (b) In this case rules ex-mov-rc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ r_i, c\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto c\}$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rule e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, c\rangle \xrightarrow{e} \langle \sigma, \pi, c\rangle$.
   We know that $\mu_a \vdash c \leftrightsquigarrow c$. Hence, the update registers are still related.
7. Case tr-mov-r0. Then $\iota = (\mathsf{mov}_4\ r_i, 0)$, $\ell = x$ and $e = 0$.
   (a) This case is not possible. Rule ex-mov-rc always applies.
   (b) In this case rules ex-mov-rc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_4\ r_i, 0\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_4 \{r_i \mapsto 0\}$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rule e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, 0\rangle \xrightarrow{e} \langle \sigma, \pi, 0\rangle$.
   We know that $\mu_a \vdash 0 \leftrightsquigarrow 0$. Hence, the update registers are still related.
8. Case tr-mov-rr. Then $\iota = (\mathsf{mov}_w\ r_i, r_j)$, $\ell = x$ and $e = y$.
   (a) This case is not possible. Rule ex-mov-rr always applies.
   (b) In this case rules ex-mov-rr is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ r_i, r_j\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}\}$ where $\vec{b} = R_{0:w}(r_j)$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-lval and l-var we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y\rangle \xrightarrow{e} \langle \sigma, \pi, v\rangle$ where $v = \sigma(a')$ and $a' = \rho(y)$.
   From rule tr-mov-rr we know $(r_j : y)_w \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b} \leftrightsquigarrow v$. Also from rule tr-mov-rr we know $(r_i : x)_w \in \mu_\Gamma$. Hence, the registers are related. After the update we can see that they are still related.
9. Case tr-mov-ri$_1$. Then $\iota = (\mathsf{mov}_w\ r_i, [r_j])$, $\ell = x$ and $e = *y$.
   (a) This case is possible iff $R(r_j) = 0$ or $R(r_j) = \bot$. Because of the related registers and, from rule tr-mov-ri$_1$, $(r_j : y)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_j) \leftrightsquigarrow \sigma(\rho(y))$. In either of the cases for $R(r_j)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y\rangle \xrightarrow{e} \mathsf{err}$.
   (b) In this case rules ex-mov-ri is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ r_i, [r_j]\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}_2\}$ where $\vec{b}_2 = H^w(\vec{b}_1)$ and $\vec{b}_1 = R_(r_j)$.
   Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-lval, l-ptr and l-var we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, *y\rangle \xrightarrow{e} \langle \sigma, \pi, v_2\rangle$ where $v_2 = \sigma(v_1)$, $v_1 = \sigma(a')$ and $a' = \rho(y)$.
   From rule tr-mov-ri$_1$ we know $(r_j : y)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_1 \leftrightsquigarrow v_1$. From related stores, we also know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. Also from rule tr-mov-ri$_1$ we know $(r_i : x)_w \in \mu_\Gamma$. Hence, the registers

are related. After the update we can see that they are still related.

10. Case tr-mov-ri$_2$. Then $\iota = (\mathsf{mov}_w\ r_i, [r_j])$, $\ell = x$ and $e = y[0]$.
    (a) This case is possible iff $R(r_j) = 0$ or $R(r_j) = \bot$. Because of the related registers and, from rule tr-mov-ri$_2$, $(r_j : y)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_j) \leftrightsquigarrow \sigma(\rho(y))$. In either of the cases for $R(r_j)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y\rangle \xrightarrow{e} \mathsf{err}$.
    (b) In this case rules ex-mov-ri is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ r_i, [r_j]\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}_2\}$ where $\vec{b}_2 = H^w(\vec{b}_1)$ and $\vec{b}_1 = R_(r_j)$.
    Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-lval, l-ar and e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y[0]\rangle \xrightarrow{e} \langle \sigma, \pi, v_2\rangle$ where $v_2 = \sigma(v_1)$, $v_1 = \sigma(a')$ and $a' = \rho(y)$.
    From rule tr-mov-ri$_2$ we know $(r_j : y)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_1 \leftrightsquigarrow v_1$. From related stores, we also know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. Also from rule tr-mov-ri$_2$ we know $(r_i : x)_w \in \mu_\Gamma$. Hence, the registers are related. After the update we can see that they are still related.
11. Case tr-mov-ri$_3$. Then $\iota = (\mathsf{mov}_w\ r_i, [r_j])$, $\ell = x$ and $e = y \to 0$.
    (a) This case is possible iff $R(r_j) = 0$ or $R(r_j) = \bot$. Because of the related registers and, from rule tr-mov-ri$_3$, $(r_j : y)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_j) \leftrightsquigarrow \sigma(\rho(y))$. In either of the cases for $R(r_j)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y\rangle \xrightarrow{e} \mathsf{err}$.
    (b) In this case rules ex-mov-ri is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ r_i, [r_j]\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}_2\}$ where $\vec{b}_2 = H^w(\vec{b}_1)$ and $\vec{b}_1 = R_(r_j)$.
    Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-lval and l-fldwe obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y \to 0\rangle \xrightarrow{e} \langle \sigma, \pi, v_2\rangle$ where $v_2 = \sigma(v_1)$, $v_1 = \sigma(a')$ and $a' = \rho(y)$.
    From rule tr-mov-ri$_3$ we know $(r_j : y)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_1 \leftrightsquigarrow v_1$. From related stores, we also know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. Also from rule tr-mov-ri$_3$ we know $(r_i : x)_w \in \mu_\Gamma$. Hence, the registers are related. After the update we can see that they are still related.
12. Case tr-mov-ir$_1$. Then $\iota = (\mathsf{mov}_w\ [r_i], r_j)$, $\ell = *x$ and $e = y$.
    (a) This case is possible iff $R(r_i) = 0$ or $R(r_i) = \bot$. Because of the related registers and, from rule tr-mov-ir$_1$, $(r_i : x)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_i) \leftrightsquigarrow \sigma(\rho(x))$. In either of the cases for $R(r_i)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \mathsf{err}$.
    (b) In this case rules ex-mov-ir is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ [r_i], r_j\rangle \xrightarrow{\iota} \langle H', R\rangle$. Here $H' = H \circ \{\vec{b}_1, \ldots, \vec{b}_1 + (w-1) \mapsto \vec{b}_2\}$ where $\vec{b}_1 = R(r_i)$ and $\vec{b} = R_{0:w}(r_j)$.
    Similarly, through rule l-ptr $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, *x\rangle \xrightarrow{\ell} \langle \sigma, \pi, v_1\rangle$ with $v_1 = \sigma(a)$ and $a = \rho(x)$. Also through rules e-lval and l-varwe obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y\rangle \xrightarrow{e} \langle \sigma, \pi, v_2\rangle$ where $v_2 = \sigma(a')$ and $a' = \rho(y)$.
    From rule tr-mov-ir$_1$ we know $(r_j : y)_w \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. From related stores, we also know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. Also from rule tr-mov-ir$_1$ we know $(r_i : x)_w \in \mu_\Gamma$. Hence, $\mu_a \vdash \vec{b}_1 \leftrightsquigarrow v_1$. Since $(x : \theta_1 *) \in \Gamma_c$, we know that v_1 is an address. Because of related heaps, we then know that $(\vec{b}_1, v_1) in \mu_a$. After the update we can see that they are still related.
13. Case tr-mov-ir$_2$. Then $\iota = (\mathsf{mov}_w\ [r_i], r_j)$, $\ell = x[0]$ and $e = y$.
    (a) This case is possible iff $R(r_i) = 0$ or $R(r_i) = \bot$. Because of the related registers and, from rule tr-mov-ir$_2$, $(r_i : x)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_i) \leftrightsquigarrow \sigma(\rho(x))$. In either of the cases for $R(r_i)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \mathsf{err}$.
    (b) In this case rules ex-mov-ir is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ [r_i], r_j\rangle \xrightarrow{\iota} \langle H', R\rangle$. Here $H' = H \circ \{\vec{b}_1, \ldots, \vec{b}_1 + (w-1) \mapsto \vec{b}_2\}$ where $\vec{b}_1 = R(r_i)$ and $\vec{b} = R_{0:w}(r_j)$.
    Similarly, through rule l-ar and e-const $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x[0]\rangle \xrightarrow{\ell} \langle \sigma, \pi, v_1\rangle$ with $v_1 = \sigma(a)$ and $a = \rho(x)$. Also through rules e-lval and l-varwe obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y\rangle \xrightarrow{e} \langle \sigma, \pi, v_2\rangle$ where $v_2 = \sigma(a')$ and $a' = \rho(y)$.
    From rule tr-mov-ir$_2$ we know $(r_j : y)_w \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. From related stores, we also know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. Also from rule tr-mov-ir$_2$ we know $(r_i : x)_w \in \mu_\Gamma$. Hence, $\mu_a \vdash \vec{b}_1 \leftrightsquigarrow v_1$. Since $(x : \theta_1[]*) \in \Gamma_c$, we know that v_1 is an address. Because of related heaps, we then know that $(\vec{b}_1, v_1) in \mu_a$. After the update we can see that they are still related.
14. Case tr-mov-ir$_3$. Then $\iota = (\mathsf{mov}_w\ [r_i], r_j)$, $\ell = x \to 0$ and $e = y$.
    (a) This case is possible iff $R(r_i) = 0$ or $R(r_i) = \bot$. Because of the related registers and, from rule tr-mov-ir$_3$, $(r_i : x)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_i) \leftrightsquigarrow \sigma(\rho(x))$. In either of the cases for $R(r_i)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \mathsf{err}$.
    (b) In this case rules ex-mov-ir is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ [r_i], r_j\rangle \xrightarrow{\iota} \langle H', R\rangle$. Here $H' = H \circ \{\vec{b}_1, \ldots, \vec{b}_1 + (w-1) \mapsto \vec{b}_2\}$ where $\vec{b}_1 = R(r_i)$ and $\vec{b} = R_{0:w}(r_j)$.
    Similarly, through rule l-fld $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \to 0\rangle \xrightarrow{\ell} \langle \sigma, \pi, v_1\rangle$ with $v_1 = \sigma(a)$ and $a = \rho(x)$. Also through rules e-lval and l-varwe obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y\rangle \xrightarrow{e} \langle \sigma, \pi, v_2\rangle$ where $v_2 = \sigma(a')$ and $a' = \rho(y)$.
    From rule tr-mov-ir$_3$ we know $(r_j : y)_w \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. From related stores, we also know $\mu_a \vdash \vec{b}_2 \leftrightsquigarrow v_2$. Also from rule tr-mov-ir$_3$ we know $(r_i : x)_w \in \mu_\Gamma$. Hence, $\mu_a \vdash \vec{b}_1 \leftrightsquigarrow v_1$. Since $(x : N*) \in \Gamma_c$, we know that v_1 is an address. Because of related heaps, we then know that $(\vec{b}_1, v_1) in \mu_a$. After the update we can see that they are still related.
15. Case tr-mov-ri+$_1$. Then $\iota = (\mathsf{mov}_w\ r_i, [r_j + c]$, $\ell = x$ and $e = y[m]$.
    (a) This case is possible iff $R(r_j) = 0$, $R(r_j) = \bot$ or $(R(r_j) + c) \notin dom(H)$. Because of the related registers and heaps, and from rule tr-mov-ri+$_1 (r_j : y)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_j) \leftrightsquigarrow \sigma(\rho(y))$. In either of the first two cases for $R(r_j)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y[m]\rangle \xrightarrow{\ell} \mathsf{err}$. In the last case, because of related heaps, it also has to be that $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y[m]\rangle \xrightarrow{\ell} \mathsf{err}$.
    (b) In this case rules ex-mov-r+ is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ r_i, [r_j + c]\rangle \xrightarrow{\iota} \langle H, R'\rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}\}$ where $\vec{b} = H^w(\vec{b}')$ and $\vec{b} = R(r_j) +_4 c$.
    Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x\rangle \xrightarrow{\ell} \langle \sigma, \pi, a\rangle$ with $a = \rho(x)$. Also through rules e-lval, l-arand e-const we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y[m]\rangle \xrightarrow{e} \langle \sigma, \pi, v\rangle$ where $v = \sigma(a'' + m)$, $a'' = \sigma(a')$ and $a' = \rho(y)$.

From rule tr-mov-ri+$_1$ we know $(r_j : y)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}' \leftrightsquigarrow a''$. From the translation rule we also have $(y : \theta[]*) \in \Gamma_c$. Because of the progress, it means that $[a'', a'' + m] \subseteq\in \pi$. Because of the related heaps and well-typed store it follows that $\mu_a \vdash \vec{b} \leftrightsquigarrow v$. Also from rule tr-mov-ri+$_1$ we know $(r_i : x)_w \in \mu_\Gamma$. After the update we can see that they are still related.

16. Case tr-mov-ri+$_2$. Then $\iota = (\mathsf{mov}_w\ r_i, [r_j + c]$, $\ell = x$ and $e = y \to m$.
    (a) This case is possible iff $R(r_j) = 0$, $R(r_j) = \bot$ or $(R(r_j) + c) \notin dom(H)$. Because of the related registers and heaps, and from rule tr-mov-ri+$_2(r_j : y)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_j) \leftrightsquigarrow \sigma(\rho(y))$. In either of the first two cases for $R(r_j)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y[m] \rangle \xrightarrow{\ell} \mathsf{err}$. In the last case, because of related heaps, it also has to be that $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y \to m \rangle \xrightarrow{\ell} \mathsf{err}$.
    (b) In this case rules ex-mov-r+ is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ r_i, [r_j + c] \rangle \xrightarrow{\iota} \langle H, R' \rangle$. Here $R' = R \circ_w \{r_i \mapsto \vec{b}\}$ where $\vec{b} = H^w(\vec{b}')$ and $\vec{b} = R(r_j) +_4 c$.
    Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \rangle \xrightarrow{\ell} \langle \sigma, \pi, a \rangle$ with $a = \rho(x)$. Also through rules e-lval and l-fld we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y \to m \rangle \xrightarrow{e} \langle \sigma, \pi, v \rangle$ where $v = \sigma(a'' + m)$, $a'' = \sigma(a')$ and $a' = \rho(y)$.
    From rule tr-mov-ri+$_2$ we know $(r_j : y)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash \vec{b}' \leftrightsquigarrow a''$. From the translation rule we also have $(y : N*) \in \Gamma_c$ and $\Sigma(N) = \langle \theta_0, \ldots, \theta_n \rangle$. Because of the progress, it means that $[a'', a'' + m] \subseteq\in \pi$. Because of the related heaps and well-typed store it follows that $\mu_a \vdash \vec{b} \leftrightsquigarrow v$. Also from rule tr-mov-ri+$_1$ we know $(r_i : x)_w \in \mu_\Gamma$. After the update we can see that they are still related.
17. Case tr-mov-i+r$_1$. Then $\iota = (\mathsf{mov}_w\ [r_i + c], r_j$, $\ell = x[m]$ and $e = y$.
    (a) This case is possible iff $R(r_i) = 0$, $R(r_i) = \bot$ or $(R(r_i) + c) \notin dom(H)$. Because of the related registers and heaps, and from rule tr-mov-i+r$_1(r_i : x)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_i) \leftrightsquigarrow \sigma(\rho(x))$. In either of the first two cases for $R(r_i)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x[m] \rangle \xrightarrow{\ell} \mathsf{err}$. In the last case, because of related heaps, it also has to be that $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x[m] \rangle \xrightarrow{\ell} \mathsf{err}$.
    (b) In this case rules ex-mov-+r is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ [r_i + c], r_j \rangle \xrightarrow{\iota} \langle H', R \rangle$. Here $H' = H \circ \{H(R(r_i)) +_4 c +_4 n \mapsto R_{n:n+1}(r_j)\}_{n=0}^{w-1}$.
    Similarly, through rule l-ar $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x[m] \rangle \xrightarrow{\ell} \langle \sigma, \pi, a \rangle$ with $a = a' + m$ and $a' = \rho(x)$. Also through rules e-lval and l-var we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y \rangle \xrightarrow{e} \langle \sigma, \pi, v \rangle$ where $v = \sigma(a'')$ and $a'' = \rho(y)$.
    From rule tr-mov-i+r$_1$ we know $(r_i : x)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash R(r_i) \leftrightsquigarrow a'$. From the translation rule we also have $(x : \theta[]*) \in \Gamma_c$. Because of the progress, it means that $[a', a' + m] \subseteq\in \pi$. Because of the related heaps and well-typed store it follows that $(R(r_i) + c, a' + m) \in \mu_a$. Also from rule tr-mov-ri+$_1$ we know $(r_j : y)_w \in \mu_\Gamma$. Hence, $\mu_a \vdash R_{0:w}(r_j) \leftrightsquigarrow v$. After the update we can see that $(R(r_i) + c)$ and $a' + m$ are still related.
18. Case tr-mov-i+r$_2$. Then $\iota = (\mathsf{mov}_w\ [r_i + c], r_j$, $\ell = x \to m$ and $e = y$.
    (a) This case is possible iff $R(r_i) = 0$, $R(r_i) = \bot$ or $(R(r_i) + c) \notin dom(H)$. Because of the related registers and heaps, and from rule tr-mov-i+r$_2(r_i : x)_4 \in \mu_\Gamma$, we have $\mu_a \vdash R(r_i) \leftrightsquigarrow \sigma(\rho(x))$. In either of the first two cases for $R(r_i)$ we also have $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \to m \rangle \xrightarrow{\ell} \mathsf{err}$. In the last case, because of related heaps, it also has to be that $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \to m \rangle \xrightarrow{\ell} \mathsf{err}$.
    (b) In this case rules ex-mov-+r is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{mov}_w\ [r_i + c], r_j \rangle \xrightarrow{\iota} \langle H', R \rangle$. Here $H' = H \circ \{H(R(r_i)) +_4 c +_4 n \mapsto R_{n:n+1}(r_j)\}_{n=0}^{w-1}$.
    Similarly, through rule l-ar $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \to m \rangle \xrightarrow{\ell} \langle \sigma, \pi, a \rangle$ with $a = a' + m$ and $a' = \rho(x)$. Also through rules e-lval and l-var we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, y \rangle \xrightarrow{e} \langle \sigma, \pi, v \rangle$ where $v = \sigma(a'')$ and $a'' = \rho(y)$.
    From rule tr-mov-i+r$_2$ we know $(r_i : x)_4 \in \mu_\Gamma$. Hence from the related registers we know $\mu_a \vdash R(r_i) \leftrightsquigarrow a'$. From the translation rule we also have $(x : N*) \in \Gamma_c$. Because of the progress, it means that $[a', a' + m] \subseteq\in \pi$. Because of the related heaps and well-typed store it follows that $(R(r_i) + c, a' + m) \in \mu_a$. Also from rule tr-mov-ri+$_1$ we know $(r_j : y)_w \in \mu_\Gamma$. Hence, $\mu_a \vdash R_{0:w}(r_j) \leftrightsquigarrow v$. After the update we can see that $(R(r_i) + c)$ and $a' + m$ are still related.
19. Case tr-alloc-r*. Then $\iota = (\mathsf{alloc}\ r_i, r_j * c$, $\ell = x$ and $e = \mathsf{new}\ \theta[y * m]$.
    (a) Rule ex-alloc-* only fails iff $R(r_j) = \bot$. Similarly, while rules l-var, e-const and e-op do not fail, rule e-ar fails iff $\sigma(\rho(y)) = \bot$. Since $(r_j : y) \in \mu_\Gamma$, both failures coincide.
    (b) This case is similar to that of tr-alloc-rc$_2$.
20. Case tr-alloc-rc$_1$. Then $\iota = (\mathsf{alloc}\ r_i, c$, $\ell = x$ and $e = \mathsf{new}\ \theta$.
    (a) Rule ex-alloc cannot fail. Similarly, rules l-var and e-new do not fail.
    (b) In this case rules ex-alloc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{alloc}\ r_i, c \rangle \xrightarrow{\iota} \langle H', R' \rangle$. Here $R' = R \circ_4 r_i \mapsto a$. Also $H' = H \circ \{a + i \mapsto \bot\}_{i=0}^{c-1}$.
    Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \rangle \xrightarrow{\ell} \langle \sigma, \pi, a' \rangle$ where $a' = \rho(x)$. Also through rule e-new we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{new}\ \ \theta \rangle \xrightarrow{e} \langle \sigma', \pi, a'' \rangle$ where $\sigma' = \sigma \circ \{a'' \mapsto \bot\}$.
    Then choose $\mu_a' = \mu_a \circ \{(a : a'')_c\}$. Since $\mu_a \vdash \bot \leftrightsquigarrow \bot$ these fresh addresses are related. Also pick $\nu_a' = \nu_a \circ \{a + i \mapsto (a, c)\}_{i=0}^{c-1}$.
21. Case tr-alloc-rc$_2$. Then $\iota = (\mathsf{alloc}\ r_i, c$, $\ell = x$ and $e = \mathsf{new\ struct}\ N$.
    (a) Rule ex-alloc cannot fail. Similarly, rules l-var and e-str do not fail.
    (b) In this case rules ex-alloc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{alloc}\ r_i, c \rangle \xrightarrow{\iota} \langle H', R' \rangle$. Here $R' = R \circ_4 r_i \mapsto a$. Also $H' = H \circ \{a + i \mapsto \bot\}_{i=0}^{c-1}$.
    Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \rangle \xrightarrow{\ell} \langle \sigma, \pi, a' \rangle$ where $a' = \rho(x)$. Also through rule e-str we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{new\ \ struct}\ \ \theta \rangle \xrightarrow{e} \langle \sigma', \pi, a'' \rangle$ where $\sigma' = \sigma \circ \{a'' + i \mapsto \bot\}_{i=0}^{n-1}$ with n is the number of fields in the struct.
    The new memory relations are straightforward.
22. Case tr-alloc-rc$_3$. Then $\iota = (\mathsf{alloc}\ r_i, c$, $\ell = x$ and $e = \mathsf{new}\ \theta[m]$.
    (a) Rule ex-alloc cannot fail. Similarly, rules l-var,e-str and e-const do not fail.
    (b) In this case rules ex-alloc is used for progress on ι: $\vec{R} \vdash \langle H, R, \mathsf{alloc}\ r_i, c \rangle \xrightarrow{\iota} \langle H', R' \rangle$. Here $R' = R \circ_4 r_i \mapsto a$. Also $H' = H \circ \{a + i \mapsto \bot\}_{i=0}^{c-1}$.

Similarly, through rule l-var $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, x \rangle \xrightarrow{\ell} \langle \sigma, \pi, a' \rangle$ where $a' = \rho(x)$. Also through rule e-ar we obtain $\Sigma; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \mathsf{new}\ \theta[m] \rangle \xrightarrow{e} \langle \sigma', \pi, a'' \rangle$ where $\sigma' = \sigma \circ \{a'' + i \mapsto \bot\}_{i=0}^{m-1}$.
The new memory relations are straightforward.

23. Case tr-call. This case follows coinductively.

Basic Blocks The two propositions for basic blocks are the following.

Proposition 15 (Preservation of Progress for Basic Blocks)**.** If

- $\mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash b \overset{b}{\rightsquigarrow} s$
- $\forall (a : l) \in \mu_\lambda : \mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash \lambda_x(a) \overset{b}{\rightsquigarrow} \lambda_c(l)$
- $\Gamma_c; \Sigma; \Psi \vdash \rho$
- $\Sigma; \Psi \vdash \sigma; \pi$
- $\mu_a; \nu_a; \pi; \vec{\rho}, \rho \vdash H \leftrightsquigarrow \sigma$
- $\mu_a; \vec{\mu_\Gamma}, \mu_\Gamma; \sigma \vdash \vec{R}, R \leftrightsquigarrow \vec{\rho}, \rho$
- $\lambda_x; \vec{R} \vdash \langle H, R, b \rangle \xrightarrow{b} \langle H', R', b' \rangle$

then

- $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, s \rangle \xrightarrow{s} \mathsf{err}$ or
- $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, s \rangle \xrightarrow{s} \langle \sigma', \pi', s' \rangle$.

Proposition 16 (Preservation of Related Memory for Basic Blocks)**.** If

- $\mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash b \overset{b}{\rightsquigarrow} s$
- $\forall (a : l) \in \mu_\lambda : \mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash \lambda_x(a) \overset{b}{\rightsquigarrow} \lambda_c(l)$
- $\Gamma_c; \Sigma; \Psi \vdash \rho$
- $\Sigma; \Psi \vdash \sigma; \pi$
- $\mu_a; \nu_a; \pi; \vec{\rho}, \rho \vdash H \leftrightsquigarrow \sigma$
- $\mu_a; \vec{\mu_\Gamma}, \mu_\Gamma; \sigma \vdash \vec{R}, R \leftrightsquigarrow \vec{\rho}, \rho$
- $\lambda_x; \vec{R} \vdash \langle H, R, b \rangle \xrightarrow{b} \langle H', R', b' \rangle$
- $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, s \rangle \xrightarrow{s} \langle \sigma', \pi', s' \rangle$

then for some $\mu'_a \supseteq \mu_a$ and $\nu'_a \supseteq \nu_a$:

- $\mu'_a; \vec{\mu_\Gamma}, \mu_\Gamma; \sigma' \vdash \vec{R}, R \leftrightsquigarrow \vec{\rho}, \rho$
- $\mu'_a; \nu'_a; \pi'; \vec{\rho}, \rho \vdash H' \leftrightsquigarrow \sigma'$

Proof 6. The proof is straightforward.

Function Definitions The two propositions for function definitions are the following.

Proposition 17 (Preservation of Progress for Function Definitions)**.** If

- $\Sigma \vdash \langle f, \vec{r_x}, \vec{r_y}, a, \lambda_x, j \rangle \rightsquigarrow f(\overrightarrow{x : \theta}) \langle \overrightarrow{y : \theta'}, l, \lambda_c, j \rangle$
- $\mu_\Gamma = \{\overrightarrow{r_x \mapsto x}, \overrightarrow{r_y \mapsto y}\}$
- $\Gamma_c = \{\overrightarrow{x : \theta}, \overrightarrow{y : \theta'}\}$
- $\Gamma_c; \Sigma; \Psi \vdash \rho$
- $\Sigma; \Psi \vdash \sigma; \pi$
- $\mu_a; \nu_a; \pi; \vec{\rho}, \rho \vdash H \leftrightsquigarrow \sigma$
- $\mu_a; \vec{\mu_\Gamma}, \mu_\Gamma; \sigma \vdash \vec{R}, R \leftrightsquigarrow \vec{\rho}, \rho$
- $\lambda_x; \vec{R} \vdash \langle H, R, \lambda_x(a) \rangle \xrightarrow{b} \langle H', R', b' \rangle$

then

- $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \lambda_c(l) \rangle \xrightarrow{s} \mathsf{err}$ or
- $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \lambda(l) \rangle \xrightarrow{s} \langle \sigma', \pi', s' \rangle$.

Proposition 18 (Preservation of Related Memory for Function Definitions)**.** If

- $\mu_\lambda; \mu_\Gamma; \Gamma_c; \Sigma \vdash b \overset{b}{\rightsquigarrow} s$
- $\mu_\Gamma = \{\overrightarrow{r_x \mapsto x}, \overrightarrow{r_y \mapsto y}\}$
- $\Gamma_c = \{\overrightarrow{x : \theta}, \overrightarrow{y : \theta'}\}$
- $\Gamma_c; \Sigma; \Psi \vdash \rho$
- $\Sigma; \Psi \vdash \sigma; \pi$
- $\mu_a; \nu_a; \pi; \vec{\rho}, \rho \vdash H \leftrightsquigarrow \sigma$
- $\mu_a; \vec{\mu_\Gamma}, \mu_\Gamma; \sigma \vdash \vec{R}, R \leftrightsquigarrow \vec{\rho}, \rho$
- $\lambda_x; \vec{R} \vdash \langle H, R, \lambda_x(a) \rangle \xrightarrow{b} \langle H', R', b' \rangle$
- $\Sigma; \lambda_c; \vec{\rho}; \rho \vdash \langle \sigma, \pi, \lambda(l) \rangle \xrightarrow{s} \langle \sigma', \pi', s' \rangle$.

then for some $\mu'_a \supseteq \mu_a$ and $\nu'_a \supseteq \nu_a$:

- $\mu'_a; \vec{\mu_\Gamma}, \mu_\Gamma; \sigma' \vdash \vec{R}, R \leftrightsquigarrow \vec{\rho}, \rho$
- $\mu'_a; \nu'_a; \pi'; \vec{\rho}, \rho \vdash H' \leftrightsquigarrow \sigma'$

Proof 7. The proof is straightforward.