Uniqueness Types for Efficient and Verifiable Aliasing-Free Embedded Systems Programming: Soundness Proof

Tuur Benoit[1] and Bart Jacobs[2]

[1] Siemens Industry Software, Belgium
tuur.benoit@siemens.com
[2] KU Leuven, Dept. of Computer Science, imec-DistriNet Research Group, Belgium
Bart.Jacobs@cs.kuleuven.be

This report provides the soundness proof that accompanies publication [1]. In this report we show that well-typed Sim programs won't get stuck. We prove safety for Sim using progress and preservation.

Definition 1 (Store Typing).
If

- $\text{dom}(\Sigma) = \text{dom}(\Delta) = \text{dom}(\sigma)$, *and*
- $\text{dom}(\Gamma) \subset \text{dom}(\sigma)$, *and*
- $\forall y \in \text{dom}(\sigma), \Gamma.\ \texttt{wf}\ \sigma_\Gamma[y]$, *and*
- $\forall y \in \text{dom}(\Sigma), m, \tau.\ \Sigma(y) = m\ \tau \iff \Gamma \vdash y : m\ \tau$, *and*
- $\forall y \in \text{dom}(\Delta), \sigma, \Gamma.\ \Delta(y) = (\texttt{RW}|\texttt{R}) \Rightarrow \sigma_\Gamma[y] = v$, *and*
- $\forall y \in \text{dom}(\Delta), \Gamma, \tau.\ \Delta(y) = \texttt{RW} \Rightarrow \Gamma \vdash y : \texttt{mut}\ \tau$, *and*
- $\forall y \in \text{dom}(\Delta), \sigma, \Gamma, \tau.\ \Delta(y) = \texttt{W} \Rightarrow (\Gamma \vdash y : \texttt{immut}\ \tau \wedge \sigma(y) = \bot) \vee (\Gamma \vdash y : \texttt{mut}\ \tau)$,

then $\Sigma; \Delta \vdash \Gamma; \sigma$.

Theorem 1 (Type Safety). *If* $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ *and* $\Sigma; \Delta \vdash \Gamma; \sigma$ *then either* $\exists v.\ e = v$, *or there is a reduction:* $\exists \sigma', e'.\ \Gamma \vdash \sigma; e \to \sigma'; e'$ *and* $\exists \Delta'', \Delta'''.\ \Sigma; \Delta'' \vdash e' : \tau \rightsquigarrow \Delta'''$ *and* $\Sigma; \Delta'' \vdash \Gamma; \sigma'$.

The type safety theorem says that well-typed expressions will not get stuck. We are able to prove this type safety theorem by using the lemmas of progress and preservation.

Proof. We prove type safety using the standard approach of progress plus preservation.

- Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
- From H1, H2 and Lemma 1 follows H3: either $\exists v.\ e = v$, or there is a reduction: $\exists \sigma', e'.\ \Gamma \vdash \sigma; e \to \sigma'; e'$ and $\exists \Delta'', \Delta'''.\ \Sigma; \Delta'' \vdash e' : \tau \rightsquigarrow \Delta'''$.
- From H1, H2, H3 and Lemma 2 follows H4: $\exists \Delta'', \Delta'''.\ \Sigma; \Delta'' \vdash e' : \tau \rightsquigarrow \Delta'''$ and $\Sigma; \Delta'' \vdash \Gamma; \sigma'$.
- From H3 and H4 follows conclusion.

Lemma 1 (Progress). *If* $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ *and* $\Sigma; \Delta \vdash \Gamma; \sigma$ *then either* $\exists v.\ e = v$*, or there is a reduction:* $\exists \sigma', e'.\ \Gamma \vdash \sigma; e \to \sigma'; e'$.

The progress lemma says that if an expression e is well-typed and the runtime configuration is consistent with the type and permission environment, then either the expression e is a value or we can take a step to an expression e'.

The progress lemma guarantees that the reduction of an expression does not get stuck because of the state of the runtime environment. That is, if an expression type-checks, the runtime environment will always be in a state where a step can be taken, unless it is a value.

Proof. By induction on the shape of e.

- Case b: b is a value.
- Case z: z is a value.
- Case `()`: `()` is a value.
- Case $e_1; e_2$:
  1. If e_1 is not a value, by IH and rule (E-Context).
  2. Otherwise by rule (E-Seq).
- Case e `as` τ:
  1. If e_1 is not a value, by IH and rule (E-Context).
  2. By (T-As)
     (a) $\Sigma; \Delta \vdash e : u\ s$
  3. By 2a, rule (E-As) applies.
- Case $u\ s\ \{\ f_1 = e_1, f_2 = e_2\ \}$:
  1. If e_1 is not a value, then by IH and rule (E-Context).
  2. If e_1 is a value and e_2 is not a value, then by IH and rule (E-Context).
  3. Otherwise: $u\ s\ \{\ f_1 = v_1, f_2 = v_2\ \}$ is a value.
- Case $g(e)$:
  1. If e is not a value, by IH and rule (E-Context).
  2. By (T-Call)
     (a) `fn` $g(x : \tau_1) \to \tau_2\ \{\ e'\ \}$
  3. If e is a value, by 2a, rule (E-FnCall) applies.
- Case `if` e_c `then` e_t `else` e_f:
  1. If e_c is not a value, then by IH and rule (E-Context).
  2. By (T-IF)
     (a) $\Sigma; \Delta \vdash e_c$: `bool`
  3. If $e_c =$ `true`, rule (E-IfTrue) applies.
  4. Otherwise, by 2a, $e_c =$ `false` and rule (E-IfFalse) applies.
- Case `while` e_c `do` e_b:
  1. Rule (E-While) applies.
- Case `let` $m\ y : u\ s$ `in` e:
  1. By (T-LetStruct)
     (a) `wf` $u\ s$
     (b) $y \notin \mathrm{dom}(\Sigma)$
  2. Assumption $\Sigma; \Delta \vdash \Gamma; \sigma$.
  3. By Definition 1

   (a) $\mathrm{dom}(\Gamma) \subset \mathrm{dom}(\Sigma)$
4. By 1b, 2 and 3a, $y \notin \mathrm{dom}(\Gamma)$
5. By 1a, and 4, rule (E-Let) applies.

– Case `let` $m\ y : u\ r$ `in` e:
1. By (W-Sc)
   (a) $\mathsf{wf}\ u\ r$
2. By (T-LetScalar)
   (a) $y \notin \mathrm{dom}(\Sigma)$
3. Assumption $\Sigma; \Delta \vdash \Gamma; \sigma$.
4. By Definition 1
   (a) $\mathrm{dom}(\Gamma) \subset \mathrm{dom}(\Sigma)$
5. By 2a, 3 and 4a, $y \notin \mathrm{dom}(\Gamma)$
6. By 1a, and 5, rule (E-Let) applies.

– Case y:
1. By (T-Eval)
   (a) $\Sigma(y) = m\ \tau$
   (b) $\Delta(y) = (\mathtt{RW}|\mathtt{R})$
2. Assumption $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. By Definition 1
   (a) $\forall y \in \mathrm{dom}(\Sigma).\ \Sigma(y) = m\ \tau \iff \Gamma \vdash y : m\ \tau$
   (b) $\forall y \in \mathrm{dom}(\Delta).\ \Delta(y) = (\mathtt{RW}|\mathtt{R}) \Rightarrow \sigma_\Gamma[y] = v$
4. By 1a, 2 and 3a, $\Gamma \vdash y : m\ \tau$
5. By 1b, 2 and 3b, $\sigma_\Gamma[y] = v$
6. By 4, and 5, either $\mathsf{u}(v) = \mathtt{uniq}$ and rule (E-EvalUnique) applies, or $\mathsf{u}(v) = \mathtt{nonuniq}$ and rule (E-EvalNonunique) applies.

– Case $y := e$:
1. If e is not a value, by IH and rule (E-Context).
2. By (T-Assign)
   (a) $\Sigma(y) = m\ \tau$
   (b) $\Delta(y) = (\mathtt{RW}|\mathtt{W})$
3. Assumption $\Sigma; \Delta \vdash \Gamma; \sigma$.
4. By Definition 1
   (a) $\forall y \in \mathrm{dom}(\Sigma).\ \Sigma(y) = m\ \tau \iff \Gamma \vdash y : m\ \tau$
   (b) $\forall y \in \mathrm{dom}(\Delta).\ \Delta(y) = \mathtt{RW} \Rightarrow \Gamma \vdash y : \mathtt{mut}\ \tau$
   (c) $\forall y \in \mathrm{dom}(\Delta).\ \Delta(y) = \mathtt{W} \Rightarrow (\Gamma \vdash y : \mathtt{immut}\ \tau \wedge \sigma(y) = \bot) \vee (\Gamma \vdash y : \mathtt{mut}\ \tau)$
   (d) $\forall y \in \mathrm{dom}(\sigma).\ \mathsf{wf}\ \sigma_\Gamma[y]$
5. By 2a, 3 and 4a, $\Gamma \vdash y : m\ \tau$
6. By 3, 4d, (W-StVal) and definition of u, $\mathsf{u}(v) = \mathsf{u}(\tau)$
7. By 2b, 3, 4b and 4c, $(\Gamma \vdash y : \mathtt{immut}\ \tau \wedge \sigma(y) = \bot) \vee (\Gamma \vdash y : \mathtt{mut}\ \tau)$
8. By 5, 6, and 7, either $\Gamma \vdash y : \mathtt{immut}\ \tau \wedge \sigma(y) = \bot$ and rule (E-AssignImmut) applies, or $\Gamma \vdash y : \mathtt{mut}\ \tau$ and rule (E-AssignMut) applies.

– Case $\mathsf{with}(\Gamma', \sigma', e)$:
1. If e is a value, by rule (E-Return).
2. Assumption $\Sigma; \Delta \vdash \Gamma; \sigma$.

3. By (T-With)
   (a) $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$
   (b) $\Sigma; \Delta \vdash \Gamma'; \sigma'$
4. By 2, 3a, and 3b, $\Gamma' \vdash \sigma'; e \to \sigma''; e'$
5. By 2 and 4, rule (E-With) applies.

– Case $\mathsf{with}'(\Gamma', \sigma', e)$:
1. If e is a value, by rule (E-With'-Value).
2. Assumption $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. By (T-With')
   (a) $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$
   (b) $\Sigma; \Delta \vdash \Gamma'; \sigma'$
4. By 2, 3a, and 3b, $\Gamma' \vdash \sigma'; e \to \sigma''; e'$
5. By 2 and 4, rule (E-With') applies.

Lemma 2 (Preservation). *If $\Gamma \vdash \sigma; e \to \sigma'; e'$ or $\Gamma \vdash \sigma; e \to_h \sigma'; e'$ and $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and $\Sigma; \Delta \vdash \Gamma; \sigma$ then there is a Δ'', Δ''' such that $\Sigma; \Delta'' \vdash e' : \tau \rightsquigarrow \Delta'''$ and $\Sigma; \Delta'' \vdash \Gamma; \sigma'$ and $\Delta''' \geq \Delta'$.*

The preservation lemma says that if an expression e is well-typed, and the run-time configuration is consistent with the type and permission environment, and we can take a step to an expression e' then there exists a type environment Σ'', a pre-state permission environment Δ'', and a post-state permission environment Δ''' under which the updated expression e' is also well typed.

The preservation lemma ensures that a reduction preserves the invariants regarding reading of uninitialized variables, writing of immutable variables and access to variables that are present in or absent from the store.

Proof. By induction on the reduction derivation with case analysis of the last rule used in the derivation.

– Case E-Seq:
1. Assume $e = v; e_2$ and $e' = e_2$ and $\sigma' = \sigma$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-Seq), follows H3: $\Sigma; \Delta \vdash v : \tau_1 \rightsquigarrow \Delta_1$ and H4: $\Sigma; \Delta_1 \vdash e_2 : \tau \rightsquigarrow \Delta'$.
4. Take $\Delta'' = \Delta$ and $\Delta''' = \Delta'$.
5. From H3 and Lemma 5, follows H5: $\Delta_1 = \Delta$.
6. Conclusion follows trivially.

– Case E-IfTrue:
1. Assume $e = \texttt{if true then } e_t \texttt{ else } e_f$ and $e' = e_t$ and $\sigma' = \sigma$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-If), follows H3: $\Sigma; \Delta \vdash \texttt{true} : \texttt{bool} \rightsquigarrow \Delta_c$ and H4: $\Sigma; \Delta_c \vdash e_t : \tau \rightsquigarrow \Delta_t$ and H5: $\Sigma; \Delta_c \vdash e_f : \tau \rightsquigarrow \Delta_f$ and H6: $\Delta_t \Cap \Delta_f = \Delta'$.
4. From H3 and Lemma 5, follows H7: $\Delta_c = \Delta$.
5. From definition of $\Cap$, follows H8: $\Delta_t \geq \Delta'$.
6. Take $\Delta'' = \Delta$ and $\Delta''' = \Delta_t$.

7. From H4, H7, and H8, follows conclusion.

– Case E-IfFalse:
1. Assume $e = \texttt{if false then } e_t \texttt{ else } e_f$ and $e' = e_f$ and $\sigma' = \sigma$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-If), follows H3: $\Sigma; \Delta \vdash \texttt{false} : \texttt{bool} \rightsquigarrow \Delta_c$ and H4: $\Sigma; \Delta_c \vdash e_t : \tau \rightsquigarrow \Delta_t$ and H5: $\Sigma; \Delta_c \vdash e_f : \tau \rightsquigarrow \Delta_f$ and H6: $\Delta_t \Cap \Delta_f = \Delta'$.
4. From H3 and Lemma 5, follows H7: $\Delta_c = \Delta$.
5. From definition of $\Cap$, follows H8: $\Delta_f \geq \Delta'$.
6. Take $\Delta'' = \Delta$ and $\Delta''' = \Delta_f$.
7. From H4, H7, and H8, follows conclusion.

– Case E-While:
1. Assume $e = \texttt{while } e_c \texttt{ do } e_b$ and $e' = \texttt{if } e_c \texttt{ then } e_b; \texttt{while } e_c \texttt{ do } e_b \texttt{ else } ()$ and $\sigma' = \sigma$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-While), follows H3: $\Sigma; \Delta \vdash e_c : \texttt{bool} \rightsquigarrow \Delta_c$ and H4: $\Sigma; \Delta_c \vdash e_b : \tau \rightsquigarrow \Delta'$ and H5: $\Delta_c \geq \Delta'$ and H6: $\Delta' \geq \Delta$.
4. By H3, H4, H5, H6, and T-If, we have $\Sigma; \Delta \vdash \texttt{if } e_c \texttt{ then } e_b; \texttt{while } e_c \texttt{ do } e_b \texttt{ else } () : \tau \rightsquigarrow \Delta'$.
5. Take $\Delta'' = \Delta$ and $\Delta''' = \Delta'$.
6. Conclusion follows trivially.

– Case E-As:
1. Assume $e = u\ s\ \{f_1 = v_1, f_2 = v_2\} \texttt{ as uniq } s$ and $e' = \texttt{uniq } s\ \{f_1 = v_1, f_2 = v_2\}$ and $\sigma' = \sigma$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-As), follows H3: $\Sigma; \Delta \vdash u\ s\ \{f_1 = v_1, f_2 = v_2\} : u\ s \rightsquigarrow \Delta'$.
4. Take $\Delta'' = \Delta$ and $\Delta''' = \Delta'$.
5. Conclusion follows trivially.

– Case E-FnCall:
1. Assume $e = g(v)$ and $e' = \mathsf{with}((\emptyset, x : \texttt{immut } \tau), \emptyset[x \mapsto v], e_0)$ and $\sigma' = \sigma$ and $\texttt{fn } g(x : \tau_0) \to \tau_0' \ \{\ e_0\ \}$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-Call), follows H3: $\Sigma; \Delta \vdash v : \tau_1 \rightsquigarrow \Delta'$.
4. From H3 and Lemma 5, follows H4: $\Delta' = \Delta$.
5. From H3 and T-With, follows $\Sigma_w; \Delta_w \vdash \mathsf{with}((\emptyset, x : \texttt{immut } \tau), \emptyset[x \mapsto v], e_0) \rightsquigarrow \Delta_w$
6. Take $\Delta'' = \Delta_w$ and $\Delta''' = \Delta_w$.
7. Conclusion follows trivially.

– Case E-With:
1. Assume $e = \mathsf{with}(\Gamma_0', \sigma_0', e_0)$ and $e' = \mathsf{with}(\Gamma_0', \sigma_0'', e_0')$ and $\sigma' = \sigma$.
2. Assume H0: $\Gamma_0' \vdash \sigma_0'; e_0 \to \sigma_0''; e_0'$.
3. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.

4. From inversion on H1 (only possible rule: T-With), follows H3: $\Sigma_0'; \Delta_0' \vdash e_0 : \tau \rightsquigarrow \Delta_0''$ and H4: $\Sigma_0'; \Delta_0' \vdash \Gamma_0'; \sigma_0'$.
5. From H0, T-With and IH, follows $\Sigma_0'; \Delta_0' \vdash \mathsf{with}(\Gamma_0', \sigma_0'', e_0') : \tau \rightsquigarrow \Delta_0''$.
6. Take $\Delta'' = \Delta_0'$ and $\Delta''' = \Delta_0''$.
7. Conclusion follows trivially.

– Case E-Return:
1. Assume $e = \mathsf{with}(\Gamma_0', \sigma_0', v)$ and $e' = v$ and $\sigma' = \sigma$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-With), follows H3: $\Sigma_0'; \Delta_0' \vdash v : \tau \rightsquigarrow \Delta_0''$ and H4: $\Sigma_0'; \Delta_0' \vdash \Gamma_0'; \sigma_0'$.
4. Take $\Delta'' = \Delta_0'$ and $\Delta''' = \Delta_0''$.
5. Conclusion follows trivially.

– Case E-Let:
1. Assume $e = \texttt{let}\ m\ x : \tau\ \texttt{in}\ e_0$ and $e' = \mathsf{with}'((\Gamma, x : m\ \tau), \sigma[x \mapsto_{(\Gamma, x:m\ \tau)} \bot], e_0)$ and $\sigma' = \sigma$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. Case $\tau = r$
   (a) From inversion on H1 (only possible rule: T-LetScalar), follows H3: $(\Sigma, x : m\ r); (\Delta, x \mapsto \mathtt{W}) \vdash e_0 : \tau \rightsquigarrow \Delta_1$ and H4: $\Delta_1|_{\mathrm{dom}(\Sigma)} = \Delta'$.
   (b) From H3, H4 and T-With', follows $\Sigma; \Delta \vdash \mathsf{with}'((\Gamma, x : m\ \tau), \sigma[x \mapsto_{(\Gamma, x:m\ \tau)} \bot], e_0) : \tau \rightsquigarrow \Delta'$.
   (c) Take $\Delta'' = \Delta$ and $\Delta''' = \Delta'$.
   (d) Conclusion follows trivially.
4. Case $\tau = u\ s$
   (a) From inversion on H1 (only possible rule: T-LetStruct), follows H5: $(\Sigma, y : m\ u\ s); (\Delta, y \mapsto \mathtt{W}) \vdash \texttt{let}\ m\ y.f_1 : \tau_1\ \texttt{in let}\ m\ y.f_2 : \tau_2\ \texttt{in}\ e : \tau \rightsquigarrow \Delta_1$ and H6: $\Delta_1|_{\mathrm{dom}(\Sigma)} = \Delta'$.
   (b) From H5, H6 and T-With', follows $\Sigma; \Delta \vdash \mathsf{with}'((\Gamma, x : m\ \tau), \sigma[x \mapsto_{(\Gamma, x:m\ \tau)} \bot], e_0) : \tau \rightsquigarrow \Delta'$.
   (c) Take $\Delta'' = \Delta$ and $\Delta''' = \Delta'$.
   (d) Conclusion follows trivially.

– Case E-With':
1. Assume $e = \mathsf{with}'(\Gamma_0', \sigma_0', e_0)$ and $e' = \mathsf{with}'(\Gamma_0', \sigma_0'', e_0')$ and $\sigma' = \sigma$.
2. Assume H0: $\Gamma_0' \vdash \sigma_0'; e_0 \to \sigma_0''; e_0'$.
3. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
4. From inversion on H1 (only possible rule: T-With'), follows H3: $\Sigma_0'; \Delta_0' \vdash e_0 : \tau \rightsquigarrow \Delta_0''$ and H4: $\Sigma_0'; \Delta_0' \vdash \Gamma_0'; \sigma_0'$.
5. From H0, T-With and IH, follows $\Sigma_0'; \Delta_0' \vdash \mathsf{with}(\Gamma_0', \sigma_0'', e_0') : \tau \rightsquigarrow \Delta_0''$.
6. Take $\Delta'' = \Delta_0'$ and $\Delta''' = \Delta_0''$.
7. Conclusion follows trivially.

– Case E-With'-Value:
1. Assume $e = \mathsf{with}'(\Gamma_0', \sigma_0', v)$ and $e' = v$ and $\sigma' = \sigma_0'|_{\mathrm{dom}(\Gamma)}$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. From inversion on H1 (only possible rule: T-With'), follows
   (a) H3: $\Sigma; \Delta \vdash \Gamma_0'|_{\mathrm{dom}(\Sigma)}; \sigma_0'|_{\mathrm{dom}(\Sigma)}$

   (b) H4: $\Sigma_0; \Delta_0 \vdash \Gamma_0'; \sigma_0'$
   (c) H5: $\Sigma_0; \Delta_0 \vdash v : \tau \rightsquigarrow \Delta_1$
   (d) H6: $\Delta' = \Delta_1|_{\mathrm{dom}(\Sigma)}$
4. Take $\Delta'' = \Delta$ and $\Delta''' = \Delta'$.
5. Conclusion follows trivially.

– Case E-AssignImmut:
1. Assume $e = y := v$ and $e' = ()$ and $\sigma' = \sigma[y \mapsto v]$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. By inversion on H1 (only possible rule: T-Assign), we have
   (a) H3: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta'_a$
   (b) H4: $\Delta'_a(y) \in \{\mathtt{RW}, \mathtt{W}\}$
   (c) H5: $\Delta' = \mathsf{norm}(\Delta'_a[y \rightrightarrows_\Sigma \rho(m)])$
4. From H3 and Lemma 5, follows H6: $\Delta'_a = \Delta$.
5. From H5 and H6 follows $\Delta' = \mathsf{norm}(\Delta[y \rightrightarrows_\Sigma \rho(m)])$
6. Take $\Delta'' = \Delta'$ and $\Delta''' = \Delta'$.
7. From Definition 1 and T-Norm follows $\Sigma; \Delta'' \vdash \Gamma; \sigma'$.
8. From T-Unit follows $\Sigma; \Delta'' \vdash () : \tau \rightsquigarrow \Delta''$
9. Conclusion follows trivially.

– Case E-AssignMut:
1. Following the same proof steps of E-AssignImmut.

– Case E-EvalUnique:
1. Assume $e = y$ and $e' = v$ and $\sigma_\gamma[y] = v$ and $\sigma' = \sigma -_\Gamma y$ and $\mathsf{u}(v) = \mathtt{uniq}$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. By inversion on H1 (only possible rule: T-Eval), we have
   (a) H3: $\Sigma(y) = m\ \tau$
   (b) H4: $\Delta(y) \in \{\mathtt{RW}, \mathtt{R}\}$
   (c) H5: $\Delta' = \mathsf{norm}(\Delta[y \rightrightarrows_\Sigma \pi(\mathsf{u}(\tau), \Delta(y))])$
4. Take $\Delta'' = \Delta'$ and $\Delta''' = \Delta'$.
5. From Definition 1 and T-Norm and definition of π follows $\Sigma; \Delta'' \vdash \Gamma; \sigma -_\Gamma y$.
6. Conclusion follows trivially.

– Case E-EvalNonunique:
1. Assume $e = y$ and $e' = v$ and $\sigma_\gamma[y] = v$ and $\sigma' = \sigma$ and $\mathsf{u}(v) = \mathtt{nonuniq}$.
2. Assume H1: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$.
3. By inversion on H1 (only possible rule: T-Eval), we have
   (a) H3: $\Sigma(y) = m\ \tau$
   (b) H4: $\Delta(y) \in \{\mathtt{RW}, \mathtt{R}\}$
   (c) H5: $\Delta' = \mathsf{norm}(\Delta[y \rightrightarrows_\Sigma \pi(\mathsf{u}(\tau), \Delta(y))])$
4. Take $\Delta'' = \Delta'$ and $\Delta''' = \Delta'$.
5. From Definition 1 and T-Norm and definition of π follows $\Sigma; \Delta'' \vdash \Gamma; \sigma$.
6. Conclusion follows trivially.

– Case E-Context:
1. Assume $e = K[e_\mathsf{h}]$ and $e' = K[e'_\mathsf{h}]$.
2. Assume H1: $\Gamma \vdash \sigma; e_\mathsf{h} \rightarrow_\mathsf{h} \sigma'; e'_\mathsf{h}$.
3. Assume H2: $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and H3: $\Sigma; \Delta \vdash \Gamma; \sigma$.
4. From IH, H1, H2, H3, and Lemma 3 follows conclusion.

Lemma 3 (Preservation Context). *If* $\Sigma; \Delta \vdash K[e_\mathsf{h}] : \tau \rightsquigarrow \Delta'$ *and* $\Sigma; \Delta \vdash \Gamma; \sigma$ *and* $\Gamma \vdash \sigma; e_\mathsf{h} \rightarrow_\mathsf{h} \sigma'; e'_\mathsf{h}$ *and there exists a* Δ''_h, Δ'''_h *such that* $\Sigma; \Delta''_h \vdash K[] : \tau \rightsquigarrow \Delta'''_h$ *and* $\Sigma; \Delta''_h \vdash \Gamma; \sigma'_h$ *and* $\Delta''' \geq \Delta'$ *,then there exists a* Δ'', Δ''' *such that* $\Sigma; \Delta'' \vdash K[e'_\mathsf{h}] : \tau \rightsquigarrow \Delta'''$ *and* $\Sigma; \Delta'' \vdash \Gamma; \sigma'$ *and* $\Delta''' \geq \Delta'$.

Proof. By induction on K.

- Case $K = K'; e_2$:
  1. Assume H1: $\Sigma; \Delta \vdash K'[e_\mathsf{h}]; e_2 : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$ and H3: $\Gamma \vdash \sigma; e_\mathsf{h} \rightarrow_\mathsf{h} \sigma'; e'_\mathsf{h}$.
  2. From inversion on H1 (only possible rule: T-Seq), follows H4: $\Sigma; \Delta \vdash K'[e_\mathsf{h}] : \tau_1 \rightsquigarrow \Delta_1$ and H5: $\Sigma; \Delta_1 \vdash e_2 : \tau \rightsquigarrow \Delta'$.
  3. From H4 and IH, follows H6: there exists a Δ''_1, Δ'''_1 for which $\Sigma; \Delta''_1 \vdash K'[e'_\mathsf{h}] : \tau_1 \rightsquigarrow \Delta'''_1$ and $\Delta'''_1 \geq \Delta_1$.
  4. From H5 and Lemma 4, follows H7: there exists a Δ'''_2 for which $\Sigma; \Delta'''_1 \vdash e_2 : \tau \rightsquigarrow \Delta'''_2$ and $\Delta'''_2 \geq \Delta'$.
  5. Take $\Delta'' = \Delta''_1$ and $\Delta''' = \Delta'''_2$.
  6. From H6, H7, and T-Seq, follows conclusion.
- Case $K = u\ s\ \{\ f_1 = K', f_2 = e_2\ \}$:
  1. Assume H1: $\Sigma; \Delta \vdash u\ s\ \{\ f_1 = K'[e_\mathsf{h}], f_2 = e_2\ \} : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$ and H3: $\Gamma \vdash \sigma; e_\mathsf{h} \rightarrow_\mathsf{h} \sigma'; e'_\mathsf{h}$.
  2. From inversion on H1 (only possible rule: T-StructExpr), follows H4: $\Sigma; \Delta \vdash K'[e_\mathsf{h}] : \tau_1 \rightsquigarrow \Delta_1$ and H5: $\Sigma; \Delta_1 \vdash e_2 : \tau \rightsquigarrow \Delta'$.
  3. From H4 and IH, follows H6: there exists a Δ''_1, Δ'''_1 for which $\Sigma; \Delta''_1 \vdash K'[e'_\mathsf{h}] : \tau_1 \rightsquigarrow \Delta'''_1$ and $\Delta'''_1 \geq \Delta_1$.
  4. From H5 and Lemma 4, follows H7: there exists a Δ'''_2 for which $\Sigma; \Delta'''_1 \vdash e_2 : \tau \rightsquigarrow \Delta'''_2$ and $\Delta'''_2 \geq \Delta'$.
  5. Take $\Delta'' = \Delta''_1$ and $\Delta''' = \Delta'''_2$.
  6. From H6, H7, and T-StructExpr, follows conclusion.
- Case $K = u\ s\ \{\ f_1 = v_1, f_2 = K'\ \}$:
  1. Assume H1: $\Sigma; \Delta \vdash u\ s\ \{\ f_1 = v_1, f_2 = K'[e_\mathsf{h}]\ \} : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$ and H3: $\Gamma \vdash \sigma; e_\mathsf{h} \rightarrow_\mathsf{h} \sigma'; e'_\mathsf{h}$.
  2. From inversion on H1 (only possible rule: T-StructExpr), follows H4: $\Sigma; \Delta \vdash v_1 : \tau_1 \rightsquigarrow \Delta_1$ and H5: $\Sigma; \Delta_1 \vdash K'[e_\mathsf{h}] : \tau \rightsquigarrow \Delta'$.
  3. From H4 and Lemma 5, follows H6: $\Delta_1 = \Delta$.
  4. From H5, H6 and IH, follows conclusion.
- Case $K = K'$ `as` τ:
  1. Assume H1: $\Sigma; \Delta \vdash K'[e_\mathsf{h}]$ `as` $\tau : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$ and H3: $\Gamma \vdash \sigma; e_\mathsf{h} \rightarrow_\mathsf{h} \sigma'; e'_\mathsf{h}$ and $\tau =$ `uniq` s.
  2. From inversion on H1 (only possible rule: T-As), follows H4: $\Sigma; \Delta \vdash K'[e_\mathsf{h}] : u\ s \rightsquigarrow \Delta'$.
  3. From H4 and IH, follows H5: there exists a Δ''_1, Δ'''_1 for which $\Sigma; \Delta''_1 \vdash K'[e'_\mathsf{h}] : u\ s \rightsquigarrow \Delta'''_1$ and $\Delta'''_1 \geq \Delta'$.
  4. Take $\Delta'' = \Delta''_1$ and $\Delta''' = \Delta'''_1$.
  5. From H5 and T-As, follows conclusion.
- Case $K = y := K'$:

1. Assume H1: $\Sigma; \Delta \vdash y := K'[e_{\mathsf{h}}] : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$ and H3: $\Gamma \vdash \sigma; e_{\mathsf{h}} \rightarrow_{\mathsf{h}} \sigma'; e'_{\mathsf{h}}$.
2. From inversion on H1 (only possible rule: T-Assign), follows H4: $\Sigma; \Delta \vdash K'[e_{\mathsf{h}}] : \tau_1 \rightsquigarrow \Delta_1$ and H5: $\Delta_1(y) \in \{\mathtt{RW}, \mathtt{W}\}$ and H6: $\Delta' = \mathsf{norm}(\Delta_1[y \rightrightarrows_\Sigma \rho(m)])$.
3. From H4 and IH, follows H7: there exists a Δ''_1, Δ'''_1 for which $\Sigma; \Delta''_1 \vdash K'[e'_{\mathsf{h}}] : \tau_1 \rightsquigarrow \Delta'''_1$ and $\Delta'''_1 \geq \Delta_1$.
4. From H7, H6 and definition of norm, follows H8: $\Delta'''_1 \geq \Delta'$.
5. Take $\Delta'' = \Delta''_1$ and $\Delta''' = \Delta'''_1$.
6. From H7, H8 and T-Assign, follows conclusion.

– Case $K = g(K')$:
1. Assume H1: $\Sigma; \Delta \vdash g(K'[e_{\mathsf{h}}]) : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$ and H3: $\Gamma \vdash \sigma; e_{\mathsf{h}} \rightarrow_{\mathsf{h}} \sigma'; e'_{\mathsf{h}}$.
2. From inversion on H1 (only possible rule: T-Call), follows H4: $\Sigma; \Delta \vdash K'[e_{\mathsf{h}}] : \tau_1 \rightsquigarrow \Delta'$.
3. From H4 and IH, follows H5: there exists a Δ''_1, Δ'''_1 for which $\Sigma; \Delta''_1 \vdash K'[e'_{\mathsf{h}}] : \tau_1 \rightsquigarrow \Delta'''_1$ and $\Delta'''_1 \geq \Delta'$.
4. Take $\Delta'' = \Delta''_1$ and $\Delta''' = \Delta'''_1$.
5. From H5 and T-Seq, follows conclusion.

– Case $K = \mathtt{if}\ K'\ \mathtt{then}\ e_t\ \mathtt{else}\ e_f$:
1. Assume H1: $\Sigma; \Delta \vdash \mathtt{if}\ K'[e_{\mathsf{h}}]\ \mathtt{then}\ e_t\ \mathtt{else}\ e_f : \tau \rightsquigarrow \Delta'$ and H2: $\Sigma; \Delta \vdash \Gamma; \sigma$ and H3: $\Gamma \vdash \sigma; e_{\mathsf{h}} \rightarrow_{\mathsf{h}} \sigma'; e'_{\mathsf{h}}$.
2. From inversion on H1 (only possible rule: T-If), follows H4: $\Sigma; \Delta \vdash K'[e_{\mathsf{h}}] : \mathtt{bool} \rightsquigarrow \Delta_c$ and H5: $\Sigma; \Delta_c \vdash e_t : \tau \rightsquigarrow \Delta_t$ and H6: $\Sigma; \Delta_c \vdash e_f : \tau \rightsquigarrow \Delta_f$ and H7: $\Delta' = \Delta_t \Cap \Delta_f$.
3. From H4 and IH, follows H8: there exists a Δ''_c, Δ'''_c for which $\Sigma; \Delta''_c \vdash K'[e'_{\mathsf{h}}] : \mathtt{bool} \rightsquigarrow \Delta'''_c$ and $\Delta'''_c \geq \Delta_c$.
4. From H5 and Lemma 4, follows H9: there exists a Δ'''_t for which $\Sigma; \Delta'''_c \vdash e_t : \tau \rightsquigarrow \Delta'''_t$ and $\Delta'''_t \geq \Delta_t$.
5. From H6 and Lemma 4, follows H10: there exists a Δ'''_f for which $\Sigma; \Delta'''_c \vdash e_f : \tau \rightsquigarrow \Delta'''_f$ and $\Delta'''_f \geq \Delta_f$.
6. Take $\Delta'' = \Delta''_c$ and $\Delta''' = \Delta'''_t \Cap \Delta'''_f$.
7. From H9, H10, and T-If, follows conclusion.

Lemma 4 (Extra). *If $\Sigma; \Delta \vdash e : \tau \rightsquigarrow \Delta'$ and $\Delta'' \geq \Delta$ then $\Sigma; \Delta'' \vdash e : \tau \rightsquigarrow \Delta' + (\Delta'' - \Delta)$.*

Proof. By induction on the type derivation with case analysis of the rule used.

– Case T-Bool:
1. Assume H1: $\Sigma; \Delta \vdash b : \mathtt{bool} \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash b : \mathtt{bool} \rightsquigarrow \Delta$.
3. From H1 and H3, follows H4: $\Delta' = \Delta$.
4. From H4 and T-Bool, conclusion follows trivially.

– Case T-Int:
1. Assume H1: $\Sigma; \Delta \vdash z : \mathtt{int} \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash z : \mathtt{int} \rightsquigarrow \Delta$.

3. From H1 and H3, follows H4: $\Delta' = \Delta$.
4. From H4 and T-Int, conclusion follows trivially.

– Case T-Unit:
1. Assume H1: $\Sigma; \Delta \vdash () : \texttt{unit} \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash () : \texttt{unit} \rightsquigarrow \Delta$.
3. From H1 and H3, follows H4: $\Delta' = \Delta$.
4. From H4 and T-Unit, conclusion follows trivially.

– Case T-Call:
1. Assume H1: $\Sigma; \Delta \vdash g(e_1) : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash e_1 : \tau_1 \rightsquigarrow \Delta'$.
3. From H3 and IH, follows H4: $\Sigma; \Delta'' \vdash e_1 : \tau_1 \rightsquigarrow \Delta' + (\Delta'' - \Delta)$
4. From H4 and T-Call follows conclusion.

– Case T-Seq:
1. Assume H1: $\Sigma; \Delta \vdash e_1; e_2 : \tau_2 \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash e_1 : \tau_1 \rightsquigarrow \Delta_1$ and H4: $\Sigma; \Delta_1 \vdash e_2 : \tau_2 \rightsquigarrow \Delta'$.
3. From H3 and IH, follows H5: $\Sigma; \Delta'' \vdash e_1 : \tau_1 \rightsquigarrow \Delta_1 + (\Delta'' - \Delta)$
4. From H4 and IH, follows H6: $\Sigma; \Delta_1 + (\Delta'' - \Delta) \vdash e_2 : \tau_2 \rightsquigarrow \Delta' + (\Delta_1 + (\Delta'' - \Delta) - \Delta_1)$
5. From H6 follows H7: $\Sigma; \Delta_1 + (\Delta'' - \Delta) \vdash e_2 : \tau_2 \rightsquigarrow \Delta' + (\Delta'' - \Delta)$
6. From H5, H7 and T-Seq follows conclusion.

– Case T-As:
1. Assume H1: $\Sigma; \Delta \vdash e_0 \texttt{ as uniq } s : \texttt{uniq } s \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash e_0 : u\ s \rightsquigarrow \Delta'$.
3. From H3 and IH, follows H4: $\Sigma; \Delta'' \vdash e_0 : u\ s \rightsquigarrow \Delta' + (\Delta'' - \Delta)$
4. From H4 and T-Call follows conclusion.

– Case T-StructExpr:
1. Assume H1: $\Sigma; \Delta \vdash u\ s\ \{\ f_1 = e_1, f_2 = e_2\ \} : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash e_1 : \tau_1 \rightsquigarrow \Delta_1$ and H4: $\Sigma; \Delta_1 \vdash e_2 : \tau_2 \rightsquigarrow \Delta'$.
3. From H3 and IH, follows H5: $\Sigma; \Delta'' \vdash e_1 : \tau_1 \rightsquigarrow \Delta_1 + (\Delta'' - \Delta)$
4. From H4 and IH, follows H6: $\Sigma; \Delta_1 + (\Delta'' - \Delta) \vdash e_2 : \tau_2 \rightsquigarrow \Delta' + (\Delta_1 + (\Delta'' - \Delta) - \Delta_1)$
5. From H6 follows H7: $\Sigma; \Delta_1 + (\Delta'' - \Delta) \vdash e_2 : \tau_2 \rightsquigarrow \Delta' + (\Delta'' - \Delta)$
6. From H5, H7 and T-StructExpr follows conclusion.

– Case T-If:
1. Assume H1: $\Sigma; \Delta \vdash \texttt{if } e_c \texttt{ then } e_t \texttt{ else } e_f : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
2. Assume H3: $\Sigma; \Delta \vdash e_c : \tau_c \rightsquigarrow \Delta_c$ and H4: $\Sigma; \Delta_c \vdash e_t : \tau \rightsquigarrow \Delta_t$ and H5: $\Sigma; \Delta_c \vdash e_f : \tau \rightsquigarrow \Delta_f$ and H6: $\Delta_t \Cap \Delta_f = \Delta'$.
3. From H3 and IH, follows H7: $\Sigma; \Delta'' \vdash e_c : \tau_c \rightsquigarrow \Delta_c + (\Delta'' - \Delta)$
4. From H4 and IH, follows H8: $\Sigma; \Delta_c + (\Delta'' - \Delta) \vdash e_t : \tau \rightsquigarrow \Delta_t + (\Delta_c + (\Delta'' - \Delta) - \Delta_c)$
5. From H8 follows H9: $\Sigma; \Delta_c + (\Delta'' - \Delta) \vdash e_t : \tau \rightsquigarrow \Delta_t + (\Delta'' - \Delta)$
6. From H5 and IH, follows H10: $\Sigma; \Delta_c + (\Delta'' - \Delta) \vdash e_f : \tau \rightsquigarrow \Delta_f + (\Delta_c + (\Delta'' - \Delta) - \Delta_c)$
7. From H10 follows H11: $\Sigma; \Delta_c + (\Delta'' - \Delta) \vdash e_f : \tau \rightsquigarrow \Delta_f + (\Delta'' - \Delta)$
8. From definition of $\Cap$ follows H12: $(\Delta_t + (\Delta'' - \Delta)) \Cap (\Delta_f + (\Delta'' - \Delta)) = (\Delta_t \Cap \Delta_f) + (\Delta'' - \Delta)$

9. From H7, H9, H11, H12 and T-StructExpr follows conclusion.

– Case T-While:
  1. Assume H1: $\Sigma; \Delta \vdash \texttt{while}\ e_c\ \texttt{do}\ e_b : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
  2. Assume H3: $\Sigma; \Delta \vdash e_c : \texttt{bool} \rightsquigarrow \Delta_c$ and H4: $\Sigma; \Delta_c \vdash e_b : \tau_b \rightsquigarrow \Delta'$ and H5: $\Delta_c \geq \Delta'$ and H6: $\Delta' \geq \Delta$.
  3. From H3 and IH, follows H7: $\Sigma; \Delta'' \vdash e_c : \texttt{bool} \rightsquigarrow \Delta_c + (\Delta'' - \Delta)$
  4. From H4 and IH, follows H8: $\Sigma; \Delta_c + (\Delta'' - \Delta) \vdash e_b : \tau_b \rightsquigarrow \Delta' + (\Delta_c + (\Delta'' - \Delta) - \Delta_c)$
  5. From H8 follows H9: $\Sigma; \Delta_c + (\Delta'' - \Delta) \vdash e_b : \tau_b \rightsquigarrow \Delta' + (\Delta'' - \Delta)$
  6. From H5 follows H10 $\Delta_c + (\Delta'' - \Delta) \geq \Delta_b + (\Delta'' - \Delta)$.
  7. From H6 follows H11 $\Delta' + (\Delta'' - \Delta) \geq \Delta + (\Delta'' - \Delta)$.
  8. From H7, H9, H10, H11 and T-While follows conclusion.
– Case T-LetScalar:
  1. Assume H1: $\Sigma; \Delta \vdash \texttt{let}\ m\ y : r\ \texttt{in}\ e_1 : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
  2. Assume H3: $(\Sigma, y : m\ r); (\Delta, y \mapsto \texttt{W}) \vdash e_1 : \tau \rightsquigarrow \Delta_1$ and H4: $\Delta' = \Delta_1|_{\mathrm{dom}(\Sigma)}$.
  3. From H3 and IH, follows H5: $(\Sigma, y : m\ r); (\Delta'', y \mapsto \texttt{W}) \vdash e_1 : \tau \rightsquigarrow \Delta_1 + (\Delta'' - \Delta)$.
  4. From H4, H5 and T-LetScalar follows conclusion.
– Case T-LetStruct:
  1. Assume H1: $\Sigma; \Delta \vdash \texttt{let}\ m\ y : u\ s\ \texttt{in}\ e_1 : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
  2. Assume H3: $(\Sigma, y : m\ u\ s); (\Delta, y \mapsto \texttt{W}) \vdash \texttt{let}\ m\ y.f_1 : \tau_1\ \texttt{in let}\ m\ y.f_2 : \tau_2\ \texttt{in}\ e_1 : \tau \rightsquigarrow \Delta_1$ and H4: $\Delta' = \Delta_1|_{\mathrm{dom}(\Sigma)}$.
  3. From H3 and IH, follows H5: $(\Sigma, y : m\ u\ s); (\Delta'', y \mapsto \texttt{W}) \vdash \texttt{let}\ m\ y.f_1 : \tau_1\ \texttt{in let}\ m\ y.f_2 : \tau_2\ \texttt{in}\ e_1 : \tau \rightsquigarrow \Delta_1 + (\Delta'' - \Delta)$.
  4. From H4, H5 and T-LetScalar follows conclusion.
– Case T-Assign:
  1. Assume H1: $\Sigma; \Delta \vdash y := e_1 : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
  2. Assume H3: $\Sigma; \Delta \vdash e_1 : \tau_1 \rightsquigarrow \Delta_1$ and H4: $\Delta' = \mathsf{norm}(\Delta_1[y \rightrightarrows_\Sigma \rho(m)])$.
  3. From H3 and IH, follows H5: $\Sigma; \Delta'' \vdash e_1 : \tau_1 \rightsquigarrow \Delta_1 + (\Delta'' - \Delta)$
  4. From definition of norm, follows H6: $\mathsf{norm}(\Delta_1 + (\Delta'' - \Delta)) = \mathsf{norm}(\Delta_1) + \mathsf{norm}(\Delta'' - \Delta)$
  5. From H4, H5, H6 and T-Assign follows conclusion.
– Case T-Eval:
  1. Assume H1: $\Sigma; \Delta \vdash y : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
  2. Assume H3: $\Delta' = \mathsf{norm}(\Delta[y \rightrightarrows_\Sigma \pi(\mathsf{u}(\tau), \Delta(y))])$.
  3. From definition of norm, follows H4: $\mathsf{norm}(\Delta' + (\Delta'' - \Delta)) = \mathsf{norm}(\Delta') + \mathsf{norm}(\Delta'' - \Delta)$
  4. From H3 and H4, follows conclusion.
– Case T-With:
  1. Assume H1: $\Sigma; \Delta \vdash \mathsf{with}(\Gamma', \sigma', e_1) : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.
  2. Assume H3: $\Sigma; \Delta \vdash \mathsf{with}(\Gamma', \sigma', e_1) : \tau \rightsquigarrow \Delta$.
  3. From H1 and H3, follows H4: $\Delta' = \Delta$.
  4. From H4 and T-With, conclusion follows trivially.
– Case T-With':
  1. Assume H1: $\Sigma; \Delta \vdash \mathsf{with}'(\Gamma', \sigma', e) : \tau \rightsquigarrow \Delta'$ and H2: $\Delta'' \geq \Delta$.

2. Assume H3: $\Sigma_0; \Delta_0 \vdash e_1 : \tau \rightsquigarrow \Delta_1$ and H4: $\Delta' = \Delta_1|_{\text{dom}(\Sigma)}$
3. From H3 and IH, follows H5: $\Sigma_0; \Delta_0'' \vdash e_1 : \tau \rightsquigarrow \Delta_1 + (\Delta_0'' - \Delta_0)$
4. How do we relate Δ_0 to Δ?

Lemma 5 (Value). *If* $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta'$ *then* $\Delta' = \Delta$.

Proof. By induction on v.

- Case $v = b$:
  1. Assume H1: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta'$.
  2. From inversion on H1 (only possible rule: T-Bool), follows H2: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta$.
  3. From H1 and H2, follows conclusion.
- Case $v = z$:
  1. Assume H1: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta'$.
  2. From inversion on H1 (only possible rule: T-Int), follows H2: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta$.
  3. From H1 and H2, follows conclusion
- Case $v = ()$:
  1. Assume H1: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta'$.
  2. From inversion on H1 (only possible rule: T-Unit), follows H2: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta$.
  3. From H1 and H2, follows conclusion
- Case $v = u\ s\ \{\ f_1 = v_1, f_2 = v_2\ \}$:
  1. Assume H1: $\Sigma; \Delta \vdash v : \tau \rightsquigarrow \Delta'$.
  2. From inversion on H1 (only possible rule: T-StructExpr), follows H2: $\Sigma; \Delta \vdash v_1 : \tau_1 \rightsquigarrow \Delta_1$ and H3: $\Sigma; \Delta_1 \vdash v_2 : \tau_2 \rightsquigarrow \Delta'$.
  3. From IH, follows conclusion.

References

1. Tuur Benoit and Bart Jacobs. Uniqueness Types for Efficient and Verifiable Aliasing-Free Embedded Systems Programming. In *Integrated Formal Methods - 15th International Conference, IFM 2019, Proceedings*, Bergen, Norway, 2019.