Refinement Types for TypeScript

– Supplemental Material –

1. Full System

In this section we present the full type system for the core language of § 3 of the main paper.

1.1 Object Constraint System

Our system leverages the idea introduced in the formall core of X10 [3] to extend a base constraint system $\mathcal{C}$ with a larger constraint system $\mathcal{O}(\mathcal{C})$, built on top of $\mathcal{C}$. The original system $\mathcal{C}$ comprises formulas taken from a decidable SMT logic [2], including, for example, linear arithmetic constraints and uninterpreted predicates. The Object Constraint System $\mathcal{O}(\mathcal{C})$ introduces the constraints:

- $\mathsf{class}(\mathsf{C})$, which it true for all classes C defined in the program;
- $x\ \mathsf{hasImm}\ \mathsf{F}$, to denote that the *immutable* field F is accessable from variable x;
- $x\ \mathsf{hasMut}\ \mathsf{G}$, to denote that the *mutable* field G is accessable from variable x; and
- $\mathsf{fields}(x) = \Diamond\overline{\mathsf{F}},\ \overline{\mathsf{G}}$, to expose all fields available to x.

Figure 1 shows the constraint system as ported from CFG [3]. We refer the reader to that work for details. The main differences are syntactic changes to account for our notion of *strengthening*. Also the FIELD rule accounts now for both immutable (as in CFJ) and mutable fields.

1.2 Well-formedness Constraints

The well-formedness rules for predicates, terms, types and heaps can be found in Figure 2. The majority of these rules are routine.

The judgment for term well-formedness assigns a *sort* to each term t, which can be thought of as a base type. The judgment $\Gamma \vdash_q \overline{t}$ is used as a shortcut for any further constraints that the f operator might impose on its arguments $\overline{t}$. For example if f is the equality operator then the two arguments are required to have types that are related via subtyping, *i.e.* if $t_1 : N_1$ and $t_2 : N_2$, it needs to be the case that $N_1 \leq N_2$ or $N_2 \leq N_1$.

Type well-formedness is typical among similar refinement types [1].

1.3 Subtyping

Figure 3 presents the full set of sybtyping rules, which borrows ideas from similar systems [1, 4].

1.4 Operational Seantics

The reduction rules for language IRSC are shown in Figure 4. These rules are re similar to the respective rules found in FCJ [3]. We use evaluation contexts E, with a left to right evaluation order, defined as:

$$\begin{aligned}\mathsf{E}\ ::=&\ \langle\ \rangle \mid \mathsf{E}.f \mid \mathsf{E}.m(\overline{u}) \mid v.m(\overline{v},\mathsf{E},\overline{u}) \mid \textbf{new}\ \mathsf{C}(\overline{v},\mathsf{E},\overline{u}) \mid \mathsf{E}\ \textbf{as}\ \mathsf{T} \mid\\ &\textbf{let}\ x = \mathsf{E}\ \textbf{in}\ u \mid \mathsf{E}.f = u \mid v.f = \mathsf{E} \mid \textbf{if}(\mathsf{E})\ \textbf{then}\ u\ \textbf{else}\ u\end{aligned}$$

$$[\textsc{Class}]\ \frac{\textbf{class}\ \mathsf{C}\,(\ldots)\ \textbf{extends}\ \mathsf{D}\ \{\ldots\} \in \overline{\mathcal{L}}}{\Gamma \vdash \mathsf{class}\,(\mathsf{C})} \qquad [\textsc{Inv}]\ \frac{\Gamma \vdash x{:}\,\mathsf{C},\ \mathsf{class}\,(\mathsf{C})}{\Gamma \vdash \mathit{inv}\,(\mathsf{C}, x)}$$

$$[\textsc{Field}]\ \frac{\Gamma \vdash \mathsf{fields}\,(x) = \Diamond\bar{f}{:}\,\overline{T},\ \bar{g}{:}\,\overline{S}}{\Gamma \vdash x\ \mathsf{hasImm}\ f_i{:}\,T_i,\ x\ \mathsf{hasMut}\ g_i{:}\,S_i} \qquad [\textsc{Object}]\ x : \mathtt{Object} \vdash \mathsf{fields}\,(x) = \varnothing$$

$$[\textsc{Field-I}]\ \frac{\Gamma, x : \mathsf{D} \vdash \mathsf{fields}\,(x) = \Diamond\bar{f}_1{:}\,\overline{T}_1,\ \bar{g}_1{:}\,\overline{S}_1 \qquad \textbf{class}\ \mathsf{C}\left(\Diamond\bar{f}_2{:}\,\overline{T}_2;\ \bar{g}_2{:}\,\overline{S}_2\right)\{p\}\ \textbf{extends}\ \mathsf{R}\ \{\ldots\} \in \overline{\mathcal{L}}}{\Gamma, x : \mathsf{D} \vdash \mathsf{fields}\,(x) = \Diamond\left(\bar{f}_1 : \overline{T}_1, \bar{f}_2 : \overline{T}_2\,[x/\mathsf{this}]\right),\ \left(\bar{g}_1 : \overline{S}_1, \bar{g}_2 : \overline{S}_2\,[x/\mathsf{this}]\right)}$$

$$[\textsc{Field-C}]\ \frac{\Gamma, x : \mathsf{C} \vdash \mathsf{fields}\,(x) = \Diamond\bar{f}{:}\,\overline{T},\ \bar{g}{:}\,\overline{S}}{\Gamma, x : \{v : \mathsf{C} \mid p\} \vdash \mathsf{fields}\,(x) = \Diamond\bar{f}{:}\,\overline{T} \Cap p\,[x/v],\ \bar{g}{:}\,\overline{S} \Cap p\,[x/v]}$$

$$[\textsc{Meth-B}]\ \frac{\Gamma \vdash \mathsf{class}\,(\mathsf{C}) \qquad \theta = [x/\mathsf{this}] \qquad \textbf{def}\ m\left(\bar{x}{:}\,\overline{T}\right)\{p\} : T = u \in \mathsf{C}}{\Gamma, x : \mathsf{C} \vdash x\ \mathsf{has}\ \left(\textbf{def}\ m\left(\bar{x}{:}\,\overline{T}\ \theta\right)\{p\ \theta\} : T\ \theta = u\right)}$$

$$[\textsc{Meth-I}]\ \frac{\Gamma, x : \mathsf{D} \vdash x\ \mathsf{has}\ \left(\textbf{def}\ m\left(\bar{x}{:}\,\overline{T}\right)\{p\} : T = u\right) \qquad \textbf{class}\ \mathsf{C}\,(\ldots)\,\{p\}\ \textbf{extends}\ \mathsf{R}\ \{\overline{\mathcal{M}}\} \in \overline{\mathcal{L}} \qquad m \notin \overline{\mathcal{M}}}{\Gamma, x : \mathsf{C} \vdash x\ \mathsf{has}\ \left(\textbf{def}\ m\left(\bar{x}{:}\,\overline{T}\right)\{p\} : T = u\right)}$$

$$[\textsc{Meth-C}]\ \frac{\Gamma, x : \mathsf{C} \vdash x\ \mathsf{has}\ \left(\textbf{def}\ m\left(\bar{x}{:}\,\overline{T}\right)\{p_0\} : T = u\right)}{\Gamma, x : \{v : \mathsf{C} \mid p\} \vdash x\ \mathsf{has}\ \left(\textbf{def}\ m\left(\bar{x}{:}\,\overline{T}\right)\{p_0\} : T \Cap [x/\mathsf{this}] = u\right)}$$

Figure 1: Structural Constraints

Well-Formed Predicates $\boxed{\Gamma \vdash p}$

$$[\textsc{WP-And}]\ \frac{\Gamma \vdash p_1 \qquad \Gamma \vdash p_2}{\Gamma \vdash p_1 \wedge p_2} \qquad [\textsc{WP-Not}]\ \frac{\Gamma \vdash p}{\Gamma \vdash \neg p} \qquad [\textsc{WP-Term}]\ \frac{\Gamma \vdash t : \mathsf{bool}}{\Gamma \vdash t}$$

Well-Formed Terms $\boxed{\Gamma \vdash t : N}$

$$[\textsc{WF-Var}]\ \frac{x : T \in \Gamma}{\Gamma \vdash x : \lfloor T \rfloor} \qquad [\textsc{WF-Const}]\ \Gamma \vdash c : \lfloor \mathsf{ty}\,(c) \rfloor \qquad [\textsc{WF-Field}]\ \frac{\Gamma \vdash t : N \qquad \Gamma, x : N \vdash x\ \mathsf{hasImm}\ f_i : T_i}{\Gamma \vdash t.f_i : \lfloor T_i \rfloor}$$

$$[\textsc{WF-Fun}]\ \frac{\Gamma \vdash f : \overline{N} \to N' \qquad \Gamma \vdash_q \bar{t}}{\Gamma \vdash f\left(\bar{t}\right) : N'}$$

Well-Formed Types $\boxed{\Gamma \vdash T}$

$$[\textsc{WT-Base}]\ \frac{\Gamma, v : N \vdash p}{\Gamma \vdash \{v : N \mid p\}} \qquad [\textsc{WT-Exists}]\ \frac{\Gamma \vdash T_1 \qquad \Gamma, x : T_1 \vdash T_2}{\Gamma \vdash \exists x{:}\,T_1.\,T_2}$$

Well-Formed Heaps $\boxed{\Gamma; \Sigma \vdash H}$

$$[\textsc{WH-Emp}]\ \Gamma; \Sigma \vdash \varnothing \qquad [\textsc{WH-Ext}]\ \frac{\Sigma[l] = T \qquad \Gamma; \Sigma \vdash o{:}\,S,\ S \leq T \qquad \Gamma; \Sigma \vdash H}{\Gamma; \Sigma \vdash l \mapsto o,\ H}$$

Figure 2: Typing Rules

Subtyping $\boxed{\Gamma \vdash T \leq T'}$

$$[\leq\text{-Refl}]\ \Gamma \vdash T \leq T \qquad [\leq\text{-Trans}]\ \frac{\Gamma \vdash T_1 \leq T_2 \qquad \Gamma \vdash T_2 \leq T_3}{\Gamma \vdash T_1 \leq T_3} \qquad [\leq\text{-Extends}]\ \frac{\textbf{class}\ C\,(\ldots)\ \textbf{extends}\ D\ \{\ldots\}}{\Gamma \vdash C \leq D}$$

$$[\leq\text{-Base}]\ \frac{\Gamma \vdash N \leq N' \qquad \mathsf{Valid}(\llbracket \Gamma \rrbracket \Rightarrow \llbracket p \rrbracket \Rightarrow \llbracket p' \rrbracket)}{\Gamma \vdash \{\nu : N \mid p\} \leq \{\nu : N' \mid p'\}} \qquad [\leq\text{-Witness}]\ \frac{\Gamma \vdash u : S \qquad \Gamma \vdash T \leq [u/x]\,T'}{\Gamma \vdash T \leq \exists x{:}\,S.\,T'}$$

$$[\leq\text{-Bind}]\ \frac{\Gamma, x : S \vdash T \leq T' \qquad x \notin FV(T')}{\Gamma \vdash \exists x{:}\,S.\,T \leq T'}$$

Figure 3: Subtyping Rules

Operational Semantics $\boxed{H, u \longmapsto H', u'}$

$$[\text{RC-ECtx}]\ \frac{H, u \longmapsto H', u'}{H, E[u] \longmapsto H', E[u']} \qquad [\text{R-Field}]\ \frac{H[l] = \textbf{new}\ C\,(\overline{v}) \qquad x : C \vdash \mathsf{fields}\,(x) = \Diamond \overline{f}{:}\,\overline{T},\ \overline{g}{:}\,\overline{S} \qquad h_i \in \overline{f} \cup \overline{g}}{H, l.h_i \longmapsto H, v_i}$$

$$[\text{R-Invk}]\ \frac{H[l] = \textbf{new}\ C\,(\ldots) \qquad x{:}\,C \vdash x\ \mathsf{has}\ \left(\textbf{def}\ m\,(\overline{x}{:}\,\overline{T})\ \{p\} : T = u\right)}{H, l.m\,(\overline{v}) \longmapsto H, [\overline{v}/\overline{x}, l/\textbf{this}]\,u} \qquad [\text{R-Cast}]\ \frac{\Gamma \vdash H[l]{:}\,S; S \leq T}{H, l\ \textbf{as}\ T \longmapsto H, l}$$

$$[\text{R-New}]\ \frac{H' = l \mapsto \textbf{new}\ C\,(\overline{v}),\ H \qquad (l\ \text{fresh})}{H, \textbf{new}\ C\,(\overline{v}) \longmapsto H', l} \qquad [\text{R-LetIn}]\ H, \textbf{let}\ x = v\ \textbf{in}\ u \longmapsto H, [v/x]\,u$$

$$[\text{RC-LetIn}]\ \frac{H, u_1 \longmapsto H', u_1'}{H, \textbf{let}\ x = u_1\ \textbf{in}\ u_2 \longmapsto H', \textbf{let}\ x = u_1'\ \textbf{in}\ u_2}$$

$$[\text{R-Asgn}]\ \frac{H[l] = \textbf{new}\ C\,(\overline{v}) \qquad H' = l \mapsto \textbf{new}\ C\,(\ldots, v_{i-1}, v, v_{i+1}, \ldots),\ H}{H, l.f_i = v \longmapsto H', v}$$

$$[\text{R-Ite-T}]\ H, \textbf{if}\,(\texttt{true})\ \textbf{then}\ u_1\ \textbf{else}\ u_2 \longmapsto H, u_1 \qquad [\text{R-Ite-F}]\ H, \textbf{if}\,(\texttt{false})\ \textbf{then}\ u_1\ \textbf{else}\ u_2 \longmapsto H, u_2$$

Figure 4: Reduction Rules

2. Proofs

Lemma 1 (Substitution Lemma). *If* $\Gamma \vdash \overline{w} : \overline{S}$*,* $\Gamma, \overline{x} : \overline{S} \vdash \overline{S} \leq \overline{S}'$*, and* $\Gamma, \overline{x} : \overline{S}' \vdash u : T$*, then* $\Gamma \vdash [\overline{w}/\overline{x}]\, u : R,\ R \leq T$.

Proof. By induction on the derivation of the statement $\Gamma, \overline{x} : \overline{S} \vdash u : T$. □

Lemma 2 (Weakening). *If* $\Gamma \vdash S \leq T$*, then* $\Gamma, x : R \vdash S \leq T$.

Proof. Straightforward. □

Lemma 3 (Store Typing Weakening). *If* $\Gamma; \Sigma \vdash u : T$*, then for some* $\Sigma' \supseteq \Sigma$*, it holds that* $\Gamma; \Sigma' \vdash u : T$.

Proof. Straightforward. □

Lemma 4 (Method Body Type – Lemma A.3 from [3]). *If*

(a) $\Gamma, z : T \vdash z \text{ has } \left(\textbf{def } m\ (\overline{z}\!: \overline{R})\ \{p\} : S = u\right)$
(b) $\Gamma, z : T, \overline{z} : \overline{T} \vdash \overline{T} \leq \overline{R}$

Then for some type S' *it is the case that:* $\Gamma, z : T, \overline{z} : \overline{T} \vdash u\!: S',\ S' \leq S$

Proof. Straightforward. □

Lemma 5 (Cast). *If* $\Gamma; \Sigma \vdash H$ *and* $\Gamma; \Sigma \vdash l : S, S \lesssim T$*, then* $\Gamma; \Sigma \vdash H[l] : R, R \leq T$

Proof. Straightforward. □

Lemma 6 (Evaluation Context Typing). *If* $\Gamma \vdash E[u] : T$*, then for some type* S *it holds that* $\Gamma \vdash u : S$*,*

Proof. By induction on the structure of the evaluation context E. □

Lemma 7 (Evaluation Context Step Typing). *If* $\Gamma; \Sigma \vdash E[u] : T, u : S$*, and for some expression* u' *and store typing* $\Sigma' \supseteq \Sigma$ *it holds that* $\Gamma; \Sigma' \vdash u'\!: S',\ S' \lesssim S$*, then* $\Gamma; \Sigma' \vdash E[u']\!: T',\ T' \lesssim T$

Proof. By induction on the structure of the evaluation context E. □

Lemma 8 (Selfification). *If* $\Gamma, x : S \vdash S \leq T$ *then* $\Gamma, x : S \vdash S \leq \text{self}\,(T, x)$.

Proof. Straightforward. □

Lemma 9 (Existential Weakening). *If* $\Gamma \vdash R \leq R'$ *then* $\Gamma \vdash \exists x\!: R.\, T \leq \exists x\!: R'.\, T$.

Proof. Straightforward. □

Lemma 10 (Existential Fold). *If* $\Gamma, z : S, x : T \vdash R \leq R'$*, then* $\Gamma, x : \exists z\!: S.\, T \vdash R \leq R'$*, where* z *does not appear in* R *and* R'.

Proof. Straightforward. □

Theorem 1 (Subject Reduction). *If*

(a) $\Gamma; \Sigma \vdash u : T$*,*
(b) $\Gamma; \Sigma \vdash H$*, and*
(c) $H, u \longmapsto H', u'$*,*

then for some T' *and* $\Sigma' \supseteq \Sigma$*:*

(d) $\Gamma; \Sigma' \vdash u' : T'$*,*
(e) $\Gamma \vdash T' \lesssim T$*, and*
(f) $\Gamma; \Sigma' \vdash H'$.

Proof. We proceed by induction on the structure of fact (c):

$$H, u \longmapsto H', u'$$

We have the following cases:

- [RC-ECtx]: Fact (c) has the form:

$$H, E[u_0] \longmapsto H', E[u_0'] \tag{6.1}$$

From (a):

$$\Gamma; \Sigma \vdash E[u_0] : T \tag{6.2}$$

From Lemma 6 on 6.2:

$$\Sigma; \Gamma \vdash u_0 : T_0 \tag{6.3}$$

By induction hypothesis, using 6.3, (b) and (c) we get:

$$\Gamma; \Sigma' \vdash u_0' : T_0' \tag{6.4}$$
$$\Gamma; \Sigma' \vdash T_0' \lesssim T_0 \tag{6.5}$$
$$\Gamma; \Sigma' \vdash H' \tag{6.6}$$
$$\Sigma' \supseteq \Sigma \tag{6.7}$$

For some type T_0' and heap H'.
From 6.6 we prove (f).
From Lemma 7 using 6.2, 6.3, 6.4, 6.5 and 6.7:

$$\Gamma; \Sigma' \vdash E[u_0'] : T', \;\; T' \lesssim T \tag{6.8}$$

From 6.8 we prove (d) and (e).

- [R-Field]: Fact (c) has the form:

$$H, l.h \longmapsto H, v \tag{6.9}$$

From (a) for $u \equiv l.h$ we have:

$$\Gamma; \Sigma \vdash l.h : T \tag{6.10}$$

By inverting R-Field on 6.9:

$$H[l] = \textbf{new}\ C\,(\overline{v}) \tag{6.11}$$

From (b) for $l \in \mathsf{dom}(H)$, it holds by WH-Ext:

$$\Gamma; \Sigma \vdash \textbf{new}\ C\,(\overline{v}) : S' \tag{6.12}$$

By inverting WH-Ext on (b):

$$\Sigma[l] = S \tag{6.13}$$
$$\Gamma \vdash S' \leq S \tag{6.14}$$

From T-New on 6.12 it holds that:

$$S' \equiv \exists \overline{z}_I{:}\overline{T}_I.\{\nu{:}C \mid \nu.\overline{f} = \overline{z}_I \wedge \mathit{inv}\,(C, \nu)\} \tag{6.15}$$

By inverting T-New on 6.12:

$$\Gamma; \Sigma \vdash \overline{v} : \left(\overline{U}_I, \overline{U}_M\right) \tag{6.16}$$
$$\vdash \mathsf{class}\,(C) \tag{6.17}$$
$$\Gamma, z{:}C; \Sigma \vdash \mathsf{fields}\,(z) = \Diamond\overline{f}{:}\overline{R},\ \overline{g}{:}\overline{V} \tag{6.18}$$
$$\Gamma, z{:}C, \overline{z}_I : \mathsf{self}\left(\overline{U}_I, z.\overline{f}\right); \Sigma \vdash \overline{U}_I \leq \overline{R},\ \overline{U}_M \leq \overline{V},\ \mathit{inv}\,(C, z) \tag{6.19}$$

We examine cases on the typing statement 6.10:

- [T-FIELD-I]: Field h is an immutable field f_i, so fact (a) becomes:

$$\Gamma; \Sigma \vdash l.f_i : \exists z\!: S.\, \mathsf{self}\,(R_i, z.f_i) \tag{6.20}$$

By inverting T-FIELD-I on 6.20:

$$\Gamma; \Sigma \vdash l : S \tag{6.21}$$
$$\Gamma, z : S; \Sigma \vdash z \ \mathsf{hasImm}\ f_i \!: R_i \tag{6.22}$$

For a fresh z.
Keeping only the relevant part of 6.16 and 6.19:

$$\Gamma; \Sigma \vdash v_i : U_i \tag{6.23}$$
$$\Gamma, z : C, \overline{z}_I : \mathsf{self}\left(\overline{U}_I, z.\overline{f}\right); \Sigma \vdash U_i \leq R_i \tag{6.24}$$

By 6.23 we prove (d).
From Lemma 8 and 6.24, picking z_i as the selfification variable:

$$\Gamma, z : C, \overline{z}_I : \mathsf{self}\left(\overline{U}_I, z.\overline{f}\right); \Sigma \vdash U_i \leq \mathsf{self}\,(R_i, z_i) \tag{6.25}$$

For the above environment it holds that:

$$[\![\Gamma, z : C, \overline{z}_I : \mathsf{self}\left(\overline{U}_I, z.\overline{f}\right); \Sigma]\!] \Rightarrow z_i = z.f_i \tag{6.26}$$

By $\leq$-REFL and From Lemma 8 using 6.26:

$$\Gamma, z : C, \overline{z}_I : \mathsf{self}\left(\overline{U}_I, z.\overline{f}\right); \Sigma \vdash \mathsf{self}\,(R_i, z_i) \leq \mathsf{self}\,(\mathsf{self}\,(R_i, z_i)\,, z.f_i) \tag{6.27}$$

By simplifying 6.27 using $\leq$-TRANS on 6.25 and 6.27 we get:

$$\Gamma, z : C, \overline{z}_I : \mathsf{self}\left(\overline{U}_I, z.\overline{f}\right); \Sigma \vdash U_i \leq \mathsf{self}\,(R_i, z.f_i) \tag{6.28}$$

From Lemma 10 using 6.15 and 6.28 we get:

$$\Gamma, z : S' \vdash U_i \leq \mathsf{self}\,(R_i, z.f_i) \tag{6.29}$$

From Rule $\leq$-WITNESS using 6.29:

$$\Gamma \vdash U_i \leq \exists z\!: S'.\, \mathsf{self}\,(R_i, z.f_i) \tag{6.30}$$

From Lemma 9 using 6.14 and 6.30:

$$\Gamma \vdash U_i \leq \exists z\!: S.\, \mathsf{self}\,(R_i, z.f_i) \tag{6.31}$$

Using 6.20, 6.16 and 6.31 we prove (e).
Heap H does not evolve so (f) holds trivially.

- [T-FIELD-M]: Field h is a mutable field g_i, so fact (a) becomes:

$$\Gamma; \Sigma \vdash l.g_i : \exists z\!: S.\, V_i \tag{6.32}$$

By inverting T-FIELD-M on 6.32:

$$\Gamma \vdash l : S \tag{6.33}$$
$$\Gamma, l : S \vdash z \ \mathsf{hasMut}\ g_i : V_i \tag{6.34}$$

For a fresh z.
Keeping only the relevant parts of 6.16 and 6.19:

$$\Gamma \vdash v_i : U_i \tag{6.35}$$
$$\Gamma, z : C, \overline{z}_I : \mathsf{self}\left(\overline{U}_I, z.\overline{f}\right) \vdash U_i \leq V_i \tag{6.36}$$

By 6.35 we prove (d).
From Lemma 10 using 6.15 and 6.36 we get:

$$\Gamma, z : S' \vdash U_i \leq V_i \tag{6.37}$$

From Rule $\leq$-WITNESS using 6.37:

$$\Gamma \vdash U_i \leq \exists z\colon S'.\, V_i \tag{6.38}$$

From Lemma 9 using 6.14 and 6.38:

$$\Gamma \vdash U_i \leq \exists z\colon S.\, V_i \tag{6.39}$$

Using 6.32, 6.16 and 6.39 we prove (e).
Heap H does not evolve so (f) holds trivially.

- [R-INVK]: Fact (c) has the form:

$$H, l.m\,(\overline{v}) \longmapsto H, [\overline{v}/\overline{z}, l/\textbf{this}]\, u' \tag{6.40}$$

From (a) for $u \equiv l.m\,(\overline{v})$ we have:

$$\Gamma; \Sigma \vdash l.m\,(\overline{v}) : \exists z\colon T.\, \exists \overline{z}\colon \overline{T}.\, S \tag{6.41}$$

By inverting T-INV on 6.41:

$$\Gamma; \Sigma \vdash l : T, \overline{v} : \overline{T} \tag{6.42}$$

$$\Gamma, z : T, \overline{z} : \overline{T} \vdash z \text{ has } \left(\textbf{def}\ m\,(\overline{z}\colon \overline{R})\ \{p\} : S = u'\right) \tag{6.43}$$

$$\Gamma, z : T, \overline{z} : \overline{T} \vdash \overline{T} \leq \overline{R} \tag{6.44}$$

$$\Gamma, z : T, \overline{z} : \overline{T} \vdash p \tag{6.45}$$

With fresh z and $\overline{z}$.
By inverting R-INVK on 6.40:

$$H[l] = \textbf{new}\ C\,(\ldots) \tag{6.46}$$

$$z\colon C \vdash z \text{ has } \left(\textbf{def}\ m\,(\overline{z}\colon \overline{R})\ \{p\} : S = u'\right) \tag{6.47}$$

Note that this has already been substituted by z in S.
By inverting WH-EXT on (c) using 6.46:

$$\Sigma[l] = T \tag{6.48}$$

$$\Gamma; \Sigma \vdash H[l]\colon T_0,\ T_0 \leq T \tag{6.49}$$

From Lemma 4 using 6.43 and 6.44:

$$\Gamma, z : T, \overline{z} : \overline{T} \vdash u'\colon S',\ S' \leq S \tag{6.50}$$

From 6.50 we prove (d).
From Rule $\leq$-WITNESS using 6.50:

$$\Gamma \vdash S' \leq \exists z\colon T.\, \exists \overline{z}\colon \overline{T}.\, S \tag{6.51}$$

From Lemma 1 using 6.42, 6.44 and 6.50:

$$\Gamma \vdash [\overline{v}/\overline{z}, l/\textbf{this}]\, u'\colon U,\ U \leq S' \tag{6.52}$$

By Rule $\leq$-TRANS on 6.50 and 6.52:

$$\Gamma \vdash U \leq \exists z\colon T.\, \exists \overline{z}\colon \overline{T}.\, S \tag{6.53}$$

From 6.53 we prove (e).
Heap H does not evolve so (f) holds trivially.

- [R-Cast]: Fact (c) has the form:

$$\mathsf{H}, \mathsf{l}\ \textbf{as}\ \mathsf{T} \longmapsto \mathsf{H}, \mathsf{l}$$

From (a) for $u \equiv l$ **as** T we have:

$$\Gamma; \Sigma \vdash l\ \textbf{as}\ T : T \tag{6.54}$$

By inverting T-Cast on 6.54:

$$\Gamma; \Sigma \vdash l : S \tag{6.55}$$

$$\Gamma \vdash T \tag{6.56}$$

$$\Gamma \vdash S \lesssim T \tag{6.57}$$

From 6.55 and 6.57 we get (d) and (e), respectively.
H does not evolve, which proves (f), given (b)

- [R-New]: Fact (c) has the form:

$$H, \textbf{new}\ C\,(\overline{v}) \longmapsto H', l$$

Where l is a fresh location and:

$$H' \equiv l \mapsto \textbf{new}\ C\,(\overline{v})\,,\ H$$

From (a) for $u \equiv$ **new** $C\,(\overline{v})$ we have:

$$\Gamma; \Sigma \vdash \textbf{new}\ C\,(\overline{v}) : R_0 \tag{6.58}$$

Where:

$$R_0 \equiv \exists \overline{z}_I{:}\,\overline{T}_I.\{v{:}\,C \mid v.\overline{f} = \overline{z}_I \wedge \mathit{inv}\,(C, v)\} \tag{6.59}$$

By inverting T-New on 6.58:

$$\Gamma \vdash \overline{v} : \left(\overline{T}_I, \overline{T}_M\right) \tag{6.60}$$

$$\vdash \mathsf{class}\,(C) \tag{6.61}$$

$$\Gamma, z : C \vdash \mathsf{fields}\,(z) = \lozenge\overline{f}{:}\,\overline{R},\ \ \overline{g}{:}\,\overline{U} \tag{6.62}$$

$$\Gamma, z : C, \overline{z} : \overline{T}, z.\overline{f} = \overline{z}_I \vdash \overline{T}_I \leq \overline{R},\ \ \overline{T}_M \leq \overline{U},\ \ \mathit{inv}\,(C, z) \tag{6.63}$$

For fresh z and $\overline{z}$.
We choose a store typing Σ', such that:

$$\Sigma' = l \mapsto R_0,\ \Sigma$$

Hence:

$$\Sigma'[l] = R_0 \tag{6.64}$$

By applying rule T-Loc using the latest equation:

$$\Gamma; \Sigma' \vdash l : R_0$$

By $\leq$-Id we trivially get:

$$\Gamma \vdash R_0 \leq R_0 \tag{6.65}$$

Which prove (d) and (e).
By applying Lemma 3 on 6.58:

$$\Gamma; \Sigma' \vdash \textbf{new}\ C\,(\overline{v}) : R_0 \tag{6.66}$$

Using 6.64, 6.65, 6.66 and (b), on rule WH-Ext:

$$\Gamma; \Sigma' \vdash H'$$

Which proves (f).

- [R-LETIN] *Similar approach to case* R-INVK.

- [R-ASGN]: Fact (c) has the form:

$$H, l.g_i = v' \longmapsto H', v' \tag{6.67}$$

By inverting R-ASGN on 6.67:

$$H[l] = \textbf{new } C(\overline{v}) \tag{6.68}$$

$$H' = l \mapsto \textbf{new } C(\ldots, v_{i-1}, v', v_{i+1}, \ldots),\ H \tag{6.69}$$

From (a) for $u \equiv l.g_i = v'$:

$$\Gamma; \Sigma \vdash l.g_i = v' : T' \tag{6.70}$$

By inverting T-ASGN on 6.70:

$$\Gamma \vdash l : T_l, v' : T' \tag{6.71}$$

$$\Gamma, z : \lfloor T_l \rfloor; \Sigma \vdash z \text{ hasMut } g_i : U_i,\ T' \leq U_i \tag{6.72}$$

With fresh z.
With 6.71 and $\leq$-REFL we prove (d) and (e).
By inverting T-LOC on 6.71:

$$\Sigma[l] = T_l \tag{6.73}$$

By inverting WH-EXT on (c) for location l, that holds an object $o \equiv H[l]$, and using 6.73:

$$\Gamma; \Sigma \vdash o : S,\ S \leq T_l \tag{6.74}$$

$$\Gamma; \Sigma \vdash H \tag{6.75}$$

By 6.68 and 6.74 we get:

$$\Gamma; \Sigma \vdash \textbf{new } C(\overline{v}) : S \tag{6.76}$$

By T-NEW, S is of the form:

$$S \equiv \exists \overline{z}_I : \overline{T}_I . \{v : C \mid v.\overline{f} = \overline{z}_I \wedge \mathit{inv}(C, v)\} \tag{6.77}$$

By inverting T-NEW on 6.76:

$$\Gamma \vdash \overline{v} : \overline{T} \tag{6.78}$$

$$\vdash \text{class}(C) \tag{6.79}$$

$$\Gamma, z : C \vdash \text{fields}(z) = \Diamond \overline{f} : \overline{R},\ \overline{g} : \overline{U} \tag{6.80}$$

$$\Gamma, z : C, \overline{z}_I : \text{self}\left(\overline{T}_I, z.\overline{f}\right) \vdash \overline{T}_I \leq \overline{R},\ \overline{T}_M \leq \overline{U},\ \mathit{inv}(C, z) \tag{6.81}$$

Where z and $\overline{z}$ are fresh and $\overline{T} \equiv \left(\overline{T}_I, \overline{T}_M\right)$.
By 6.74 it holds that:

$$\Gamma \vdash \lfloor S \rfloor \leq \lfloor T_l \rfloor \tag{6.82}$$

By 6.82 and 6.77:

$$\Gamma \vdash C \leq \lfloor T_l \rfloor \tag{6.83}$$

From Lemma A.6 in [3] using 6.72 and 6.83:

$$\Gamma, z : C \vdash T' \leq U_i \tag{6.84}$$

From Lemma 2 on 6.84:

$$\Gamma, z : C, \bar{z}_I : \mathsf{self}\left(\bar{T}_I, z.\bar{f}\right) \vdash T' \leq U_i \tag{6.85}$$

Let $\bar{z}_{M,..i-1}$ and $\bar{z}_{M,i+1,..}$, such that:

$$\bar{z}_M = \bar{z}_{M,..i-1}, z_{M,i}, \bar{z}_{M,i+1..}$$

and

$$\bar{z}'_M = \bar{z}_{M,..i-1}, z'_{M,i}, \bar{z}_{M,i+1..}$$

Also if:

$$\bar{v} = \ldots, v_{i-1}, v, v_{i+1} \ldots \quad \text{and} \quad \bar{T} = \ldots, T_{i-1}, T, T_{i+1}, \ldots$$

Then:

$$\bar{v}' = \ldots, v_{i-1}, v', v_{i+1} \ldots \quad \text{and} \quad \bar{T}' = \ldots, T_{i-1}, T', T_{i+2}, \ldots$$

Combining 6.81 and 6.85:

$$\Gamma, z : C, \bar{z}_I : \mathsf{self}\left(\bar{T}_I, z.\bar{f}\right) \vdash \bar{T}' \leq \left(\bar{R}, \bar{U}\right),\ \mathit{inv}\left(C, z\right) \tag{6.86}$$

Also from 6.71 and 6.78:

$$\Gamma \vdash \bar{v}' : \bar{T}' \tag{6.87}$$

By applying rule T-NEW using 6.87, 6.79, 6.80 and 6.86:

$$\Gamma; \Sigma \vdash \textbf{new}\ C\left(\bar{v}'\right) : S' \tag{6.88}$$

Where:

$$S' \equiv \exists \bar{z}_I{:}\bar{T}_I.\{v{:}C \mid v.\bar{f} = \bar{z}_I \wedge \mathit{inv}\left(C, v\right)\} \tag{6.89}$$

From 6.77 and 6.89:

$$S = S'$$

Also by 6.74 for $o' = \textbf{new}\ C\left(\bar{v}'\right)$:

$$\Gamma; \Sigma \vdash o' {:} S',\ S' \leq T_l \tag{6.90}$$

By applying rule WH-EXT on 6.73 6.90 and 6.75:

$$\Gamma; \Sigma \vdash l \mapsto o',\ H$$

Which proves (f).

- [R-ITE-T] *Similar approach to case* RC-ECTX.

- [R-ITE-F] *Similar approach to case* RC-ECTX.

□

Theorem 2 (Progress). *If*

(a) $\Gamma; \Sigma \vdash u : T$,
(b) $\Gamma; \Sigma \vdash H$

then one of the following holds:

- u *is a value,*
- *there exist* u', H' *and* $\Sigma' \supseteq \Sigma$ *s.t.* $\Gamma; \Sigma' \vdash H'$ *and* $H, u \longmapsto H', u'$.

Proof. We proceed by induction on the structure of the derivation: $\Gamma; \Sigma \vdash u : T$.

- [T-FIELD-I]

$$\Gamma; \Sigma \vdash u_0.f_i : \exists z{:}\, T_0.\, \mathsf{self}\,(T, z.f_i) \tag{2.1}$$

By inverting T-FIELD-I on 2.1:

$$\Gamma; \Sigma \vdash u_0 : T_0 \tag{2.2}$$
$$\Gamma, z : T_0; \Sigma \vdash z \;\mathsf{hasImm}\; f_i{:}\, T \tag{2.3}$$

By i.h. using 2.2 and (b) there are two possible cases on u_0:

  - [$u_0 \equiv l_0$] Statement 2.2 becomes:

$$\Gamma; \Sigma \vdash l_0 : T_0 \tag{2.4}$$

From (b) for location l_0:

$$\Gamma; \Sigma \vdash l_0 \mapsto o,\;\; H \tag{2.5}$$

Where:

$$o \equiv \mathbf{new}\; C\,(\overline{v}) \tag{2.6}$$

By inverting WH-EXT on 2.5:

$$\Sigma[l_0] = T_0 \tag{2.7}$$
$$\Gamma; \Sigma \vdash o{:}\, S_0,\;\; S_0 \leq T_0 \tag{2.8}$$
$$\Gamma; \Sigma \vdash H \tag{2.9}$$

From Lemma 5 using (b) and 2.8:

$$\Gamma; \Sigma \vdash o{:}\, S_0,\;\; S_0 \leq T_0 \tag{2.10}$$

From Lemma A.6 in [3] using 2.3 and 2.10:

$$\Gamma, z : S_0; \Sigma \vdash z \;\mathsf{hasImm}\; f_i{:}\, T \tag{2.11}$$

From R-FIELD using 2.5, 2.6 and 2.11:

$$H, l_0.f_i \longmapsto H, v_i$$

  - [$\exists u_0'$ *s.t.* $H, u_0 \longmapsto H', u_0'$] By rule RC-ECTX:

$$H, u_0.f_i \longmapsto H', u_0'.f_i$$

- [T-FIELD-M] *Similar to previous case.*

- [T-INV], [T-NEW] *Similar to the respective case of CFJ [3].*

- [T-CAST]:

$$\Gamma; \Sigma \vdash u_0 \;\mathbf{as}\; T : T \tag{2.12}$$

By inverting T-CAST on 2.12:

$$\Gamma \vdash u_0 : S_0 \tag{2.13}$$
$$\Gamma; \Sigma \vdash T \tag{2.14}$$
$$\Gamma; \Sigma \vdash S_0 \lesssim T \tag{2.15}$$

By i.h. using 2.13 and (b) there are two possible cases on u_0:

- $[u_0 \equiv l_0]$ Statement 2.13 becomes:

$$\Gamma; \Sigma \vdash l_0 : S_0 \tag{2.16}$$

From Lemma 5 using (b) and 2.15:

$$\Gamma; \Sigma \vdash H[l_0] \colon R_0, R_0 \leq T \tag{2.17}$$

From R-CAST using 2.17:

$$H, l_0 \textbf{ as } T \longmapsto H, l_0$$

- $[\exists u_0'$ *s.t.* $H, u_0 \longmapsto H', u_0']$ By rule RC-ECTX:

$$H, u_0 \textbf{ as } T \longmapsto H', u_0' \textbf{ as } T$$

- [T-LET], [T-ASGN], [T-IF] *These cases are handled in a similar manner.*

□

References

[1] K. Knowles and C. Flanagan. Compositional reasoning and decidable checking for dependent contract types. In *Proceedings of the 3rd Workshop on Programming Languages Meets Program Verification*, PLPV '09, pages 27–38, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-330-3.

[2] G. Nelson. Techniques for program verification. Technical Report CSL81-10, Xerox Palo Alto Research Center, 1981.

[3] N. Nystrom, V. Saraswat, J. Palsberg, and C. Grothoff. Constrained Types for Object-oriented Languages. In *Proceedings of the 23rd ACM SIGPLAN Conference on Object-oriented Programming Systems Languages and Applications*, OOPSLA '08, pages 457–474, New York, NY, USA, 2008. ACM.

[4] P. M. Rondon, M. Kawaguci, and R. Jhala. Liquid Types. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2008.