A separation logic for refining concurrent objects Technical appendix

Aaron Turon and Mitchell Wand

July 28, 2010

1 Errata

The definition of fenced projection in the paper should have said:

$$t \uplus u \in I, \rho \triangleright [\![\varphi]\!]^\rho$$

where

$$I, \rho \triangleright (\sigma, o); t = \begin{cases} (\sigma, \lightning) & \sigma, \rho \models I,\ o, \rho \not\models I \\ (\sigma, o); (I, \rho \triangleright t) & \text{otherwise} \end{cases}$$

2 Preliminaries

Definition 1. The *outcome* of κ, written $|\kappa|$, is defined by:

$$\begin{aligned} |\varphi, \sigma| &\triangleq \sigma \\ |o| &\triangleq o \end{aligned}$$

Lemma 1. $\dagger$ is a closure operator: idempotent, increasing, and monotonic.

Proof. Each property shown by straightforward induction on $\dagger$ rules. □

Lemma 2. $([\![\varphi]\!]^\rho)^\dagger = [\![\varphi]\!]^\rho$.

Proof. Straightforward induction on φ, using Lemma 1. □

Trace interleaving $s \in t \parallel u$

$$\frac{s \in t \parallel u}{s \in u \parallel t} \qquad \frac{s \in t \parallel u}{(\sigma, \sigma')s \in (\sigma, \sigma')t \parallel u} \qquad \frac{}{(\sigma, \sigma')u \in (\sigma, \sigma') \parallel u} \qquad \frac{}{(\sigma, \lightning) \in (\sigma, \lightning) \parallel u}$$

Trace set interleaving

$$T \parallel U \triangleq \{s \in t \parallel u \ :\ t \in T, u \in U\}$$

Lemma 3. $T \parallel U = U \parallel T$.

Proof. Immediate. □

Lemma 4. If T is local then $T^\dagger$ is local.

Proof. We prove pointwise: if $t \in T^\dagger$ then each trace required for locality is also in $T^\dagger$, by induction on the derivation of $t \in T^\dagger$. □

Lemma 5. If T and U are local then $T; U$ is local.

Proof. Immediate. □

Lemma 6. If T and U are local then $T \parallel U$ is local.

Proof. Pointwise, by induction on the derivation of interleaving. □

Corollary 1. $[\![\varphi]\!]^\rho$ is local.

Proof. Straightforward induction on φ, using Lemmas 5, 6 and 4. □

3 Adequacy

3.1 Raw semantics

To prove adequacy, we relate unquotiented ("raw") versions of the operational and denotational semantics, then relate the quotiented versions. The raw semantics avoid the need for stuttering or mumbling in proving adequacy. The raw semantics is given in Figure 1.

Lemma 7. $\mathcal{R}[\![\varphi]\!]^-$ is continuous.

Proof. Straightforward induction on φ, using that ;, $\parallel$, and $\cup$ preserve continuity. □

Lemma 8. $(\lightning, \lightning) \in \mathcal{R}[\![\varphi]\!]^\rho$

Proof. Straightforward induction on φ. □

Lemma 9 (Substitution)**.** If ψ is closed then $\mathcal{R}[\![\varphi]\!]^{\rho[X \mapsto \psi]} = \mathcal{R}[\![\varphi[\psi/X]]\!]^\rho$.

Proof. Straightforward induction on φ. □

Lemma 10 (Bistrength)**.** We have

1. $T^\dagger; U^\dagger \subseteq (T; U)^\dagger$
2. $T^\dagger \parallel U^\dagger \subseteq (T \parallel U)^\dagger$
3. $\bigcup(T_i^\dagger) \subseteq (\bigcup T_i)^\dagger$

Raw denotational semantics $\mathcal{R}[\![\varphi]\!]^\rho \subseteq \textsc{Trace}$

$$
\begin{aligned}
\mathcal{R}[\![\varphi;\psi]\!]^\rho &\triangleq \mathcal{R}[\![\varphi]\!]^\rho ; \mathcal{R}[\![\psi]\!]^\rho \\
\mathcal{R}[\![\varphi \parallel \psi]\!]^\rho &\triangleq \mathcal{R}[\![\varphi]\!]^\rho \parallel \mathcal{R}[\![\psi]\!]^\rho \\
\mathcal{R}[\![\varphi \vee \psi]\!]^\rho &\triangleq \mathcal{R}[\![\varphi]\!]^\rho \cup \mathcal{R}[\![\psi]\!]^\rho \\
\mathcal{R}[\![\exists x.\varphi]\!]^\rho &\triangleq \textstyle\bigcup_v \mathcal{R}[\![\varphi]\!]^{\rho[x\mapsto v]} \\
\mathcal{R}[\![\textsf{let } f = F \textsf{ in } \psi]\!]^\rho &\triangleq \mathcal{R}[\![\psi]\!]^{\rho[f\mapsto \mathcal{R}[\![F]\!]^\rho]} \\
\mathcal{R}[\![F(e)]\!]^\rho &\triangleq \mathcal{R}[\![F]\!]^\rho\,(\mathcal{R}[\![e]\!]^\rho) \\
\mathcal{R}[\![\mu X.\varphi]\!]^\rho &\triangleq \textstyle\bigcap\{T : \mathcal{R}[\![\varphi]\!]^{\rho[X\mapsto T]} \subseteq T\} \\
\mathcal{R}[\![X]\!]^\rho &\triangleq \rho(X) \\
\mathcal{R}[\![\langle \forall \overline{x} : p,\ q\rangle]\!]^\rho &\triangleq \mathrm{act}(\rho(p),\rho(q)) \cup \{(\lightning,\lightning)\} \\
\mathcal{R}[\![\{p\}]\!]^\rho &\triangleq \{(\sigma,\sigma)\ :\ \sigma \in \mathcal{R}[\![p * \textsf{true}]\!]^\rho\} \\
&\cup\ \{(\sigma,\lightning)\ :\ \sigma \notin \mathcal{R}[\![p * \textsf{true}]\!]^\rho\}
\end{aligned}
$$

Procedures:

$$\mathcal{R}[\![f]\!]^\rho \triangleq \rho(f) \qquad \mathcal{R}[\![\lambda x.\varphi]\!]^\rho \triangleq \lambda v.\mathcal{R}[\![\varphi]\!]^{\rho[x\mapsto v]}$$

Raw operational semantics $\kappa \rightsquigarrow \kappa'$

$$\frac{\varphi_1,\sigma \rightsquigarrow \varphi_1',\sigma'}{\varphi_1;\varphi_2,\sigma \rightsquigarrow \varphi_1';\varphi_2,\sigma'} \qquad \frac{\varphi_1,\sigma \rightsquigarrow \sigma'}{\varphi_1;\varphi_2,\sigma \rightsquigarrow \varphi_2,\sigma'} \qquad \frac{\varphi_1,\sigma \rightsquigarrow \lightning}{\varphi_1;\varphi_2,\sigma \rightsquigarrow \lightning}$$

$$\frac{\varphi_1 \parallel \varphi_2,\sigma \rightsquigarrow \kappa}{\varphi_2 \parallel \varphi_1,\sigma \rightsquigarrow \kappa} \qquad \frac{\varphi_1,\sigma \rightsquigarrow \varphi_1',\sigma'}{\varphi_1 \parallel \varphi_2,\sigma \rightsquigarrow \varphi_1' \parallel \varphi_2,\sigma'} \qquad \frac{\varphi_1,\sigma \rightsquigarrow \sigma'}{\varphi_1 \parallel \varphi_2,\sigma \rightsquigarrow \varphi_2,\sigma'} \qquad \frac{\varphi_1,\sigma \rightsquigarrow \lightning}{\varphi_1 \parallel \varphi_2,\sigma \rightsquigarrow \lightning}$$

$$\frac{\varphi_i,\sigma \rightsquigarrow \kappa}{\varphi_1 \vee \varphi_2,\sigma \rightsquigarrow \kappa} \qquad \frac{v \in \textsc{Val} \qquad \varphi[v/x],\sigma \rightsquigarrow \kappa}{\exists x.\varphi,\sigma \rightsquigarrow \kappa} \qquad \frac{\varphi[\mu X.\varphi/X],\sigma \rightsquigarrow \kappa}{\mu X.\varphi,\sigma \rightsquigarrow \kappa}$$

$$\frac{\varphi[F/f],\sigma \rightsquigarrow \kappa}{\textsf{let } f = F \textsf{ in } \varphi,\sigma \rightsquigarrow \kappa} \qquad \frac{\varphi[[\![e]\!]\,/x],\sigma \rightsquigarrow \kappa}{(\lambda x.\varphi)e,\sigma \rightsquigarrow \kappa}$$

$$\frac{(\sigma,o) \in \mathrm{act}(p,q)}{\langle \forall \overline{x} : p,\ q\rangle,\sigma \rightsquigarrow o} \qquad \frac{\sigma \models p * \textsf{true}}{\{p\},\sigma \rightsquigarrow \sigma} \qquad \frac{\sigma \not\models p * \textsf{true}}{\{p\},\sigma \rightsquigarrow \lightning}$$

One-step observed traces $t \in \mathcal{O}_1[\![\varphi]\!]$

$$\frac{}{(\lightning,\lightning) \in \mathcal{O}_1[\![\varphi]\!]} \qquad \frac{\varphi,\sigma \rightsquigarrow o}{(\sigma,o) \in \mathcal{O}_1[\![\varphi]\!]} \qquad \frac{\varphi,\sigma \rightsquigarrow \varphi',\sigma' \quad t \in \mathcal{O}_1[\![\varphi']\!]}{(\sigma,\sigma')t \in \mathcal{O}_1[\![\varphi]\!]}$$

Figure 1: Unquotiented semantics

Proof. For (1), we first show that $T^\dagger; U \subseteq (T;U)^\dagger$ and $T^\dagger; U^\dagger \subseteq (T;U)^\dagger$ by easy induction on the derivation $t \in T^\dagger$ or $u \in U^\dagger$ respectively. It follows that

$$T^\dagger; U^\dagger \subseteq (T; U^\dagger)^\dagger \subseteq ((T;U)^\dagger)^\dagger = (T;U)^\dagger$$

where the last step is by Lemma 1.

Parts (2) and (3) work similarly, but because $\|$ and $\cup$ are symmetric we only need to show *e.g.* $T^\dagger \parallel U \subseteq (T \parallel U)^\dagger$. □

Lemma 11. $[\![\varphi]\!]^\rho = (\mathcal{R}[\![\varphi]\!]^\rho)^\dagger$.

Proof. We prove each direction using a separate, but easy, induction on φ.

In the left-to-right direction, we appeal to Lemma 10 to push † out to the top level. For the recursive case, we use Lemma 7 and Kleene's fixpoint theorem to turn the definition (by $\cap$) into Kleene's fixpoint, which is given by $\cup$, and to which Lemma 10 applies.

To show the right-to-left direction, we show instead that $\mathcal{R}[\![\varphi]\!]^\rho \subseteq [\![\varphi]\!]^\rho$, which is a very easy induction using Lemma 1. Again by Lemma 1, it follows that $(\mathcal{R}[\![\varphi]\!]^\rho)^\dagger \subseteq ([\![\varphi]\!]^\rho)^\dagger$. Finally, since $[\![\varphi]\!]^\rho$ is †-closed (Lemma 2) the desired result follows by Lemma 1. □

3.2 Operational traces are denotational traces

Lemma 12. We have:

- If $\varphi, \sigma \rightsquigarrow \varphi', \sigma'$ and $t \in \mathcal{R}[\![\varphi']\!]$ then $(\sigma, \sigma')t \in \mathcal{R}[\![\varphi]\!]$.
- If $\varphi, \sigma \rightsquigarrow o$ then $(\sigma, o) \in \mathcal{R}[\![\varphi]\!]$.

Proof. By induction on derivations $\varphi, \sigma \rightsquigarrow \kappa$.

Case: $\boxed{\dfrac{\varphi_1, \sigma \rightsquigarrow \varphi_1', \sigma'}{\varphi_1; \varphi_2, \sigma \rightsquigarrow \varphi_1'; \varphi_2, \sigma'}}$

Let $s \in \mathcal{R}[\![\varphi_1'; \varphi_2]\!]$
$s = t; u$, $t \in \mathcal{R}[\![\varphi_1']\!]$, $u \in \mathcal{R}[\![\varphi_2]\!]$ defn
$(\sigma, \sigma')t \in \mathcal{R}[\![\varphi_1]\!]$ induction
$(\sigma, \sigma')s = (\sigma, \sigma')tu \in \mathcal{R}[\![\varphi_1; \varphi_2]\!]$ defn

Case: $\boxed{\dfrac{\varphi_1, \sigma \rightsquigarrow \sigma'}{\varphi_1; \varphi_2, \sigma \rightsquigarrow \varphi_2, \sigma'}}$

Let $s \in \mathcal{R}[\![\varphi_2]\!]$
$(\sigma, \sigma') \in \mathcal{R}[\![\varphi_1]\!]$ induction
$(\sigma, \sigma')s \in \mathcal{R}[\![\varphi_1; \varphi_2]\!]$ defn

Case: $\boxed{\dfrac{\varphi_1, \sigma \rightsquigarrow \lightning}{\varphi_1; \varphi_2, \sigma \rightsquigarrow \lightning}}$

$(\sigma, \lightning) \in \mathcal{R}[\![\varphi_1]\!]$	induction
$(\lightning, \lightning) \in \mathcal{R}[\![\varphi_2]\!]$	Lem 8
$(\sigma, \lightning) = (\sigma, \lightning); (\lightning, \lightning) \in \mathcal{R}[\![\varphi_1; \varphi_2]\!]$	defn

Case: $\boxed{\dfrac{\varphi_1 \parallel \varphi_2, \sigma \rightsquigarrow \kappa}{\varphi_2 \parallel \varphi_1, \sigma \rightsquigarrow \kappa}}$

By induction and commutativity of $\parallel$ (Lem 3).

Case: $\boxed{\dfrac{\varphi_1, \sigma \rightsquigarrow \varphi_1', \sigma'}{\varphi_1 \parallel \varphi_2, \sigma \rightsquigarrow \varphi_1' \parallel \varphi_2, \sigma'}}$

Let $s \in \mathcal{R}[\![\varphi_1'; \varphi_2]\!]$

$s \in t \parallel u$, $t \in \mathcal{R}[\![\varphi_1']\!]$, $u \in \mathcal{R}[\![\varphi_2]\!]$	defn
$(\sigma, \sigma')t \in \mathcal{R}[\![\varphi_1]\!]$	induction
$(\sigma, \sigma')s \in (\sigma, \sigma')t \parallel u \subseteq \mathcal{R}[\![\varphi_1; \varphi_2]\!]$	defn

Case: $\boxed{\dfrac{\varphi_1, \sigma \rightsquigarrow \sigma'}{\varphi_1 \parallel \varphi_2, \sigma \rightsquigarrow \varphi_2, \sigma'}}$

Let $s \in \mathcal{R}[\![\varphi_2]\!]$

$(\sigma, \sigma') \in \mathcal{R}[\![\varphi_1]\!]$	induction
$(\sigma, \sigma')s \in \mathcal{R}[\![\varphi_1 \parallel \varphi_2]\!]$	defn

Case: $\boxed{\dfrac{\varphi_1, \sigma \rightsquigarrow \lightning}{\varphi_1 \parallel \varphi_2, \sigma \rightsquigarrow \lightning}}$

$(\sigma, \lightning) \in \mathcal{R}[\![\varphi_1]\!]$	induction
$(\lightning, \lightning) \in \mathcal{R}[\![\varphi_2]\!]$	Lem 8
$(\sigma, \lightning) \in (\sigma, \lightning) \parallel (\lightning, \lightning) \subseteq \mathcal{R}[\![\varphi_1 \parallel \varphi_2]\!]$	defn

Case: $\boxed{\dfrac{\varphi_i, \sigma \rightsquigarrow \kappa}{\varphi_1 \vee \varphi_2, \sigma \rightsquigarrow \kappa}}$

By induction.

Case: $\boxed{\dfrac{v \in \textsc{Val} \qquad \varphi[v/x], \sigma \rightsquigarrow \kappa}{\exists x.\varphi, \sigma \rightsquigarrow \kappa}}$

By induction.

Case: $\boxed{\dfrac{\varphi[\mu X.\varphi/X], \sigma \rightsquigarrow \kappa}{\mu X.\varphi, \sigma \rightsquigarrow \kappa}}$

By induction and the fact that μ is denotationally a fixpoint.

Case: $\boxed{\dfrac{\varphi[F/f], \sigma \rightsquigarrow \kappa}{\textsf{let } f = F \textsf{ in } \varphi, \sigma \rightsquigarrow \kappa}}$

By induction and the substitution lemma

Case: $\boxed{\dfrac{\varphi[[\![e]\!]/x], \sigma \rightsquigarrow \kappa}{(\lambda x.\varphi)e, \sigma \rightsquigarrow \kappa}}$

By induction and the substitution lemma.

Case: $\boxed{\dfrac{(\sigma, o) \in \mathrm{act}(p, q)}{\langle \forall \overline{x} : p,\ q \rangle, \sigma \rightsquigarrow o}}$

Immediate.

Case: $\boxed{\dfrac{\sigma \models p * \textsf{true}}{\{p\}, \sigma \rightsquigarrow \sigma}}$

Immediate.

Case: $\boxed{\dfrac{\sigma \not\models p * \textsf{true}}{\{p\}, \sigma \rightsquigarrow \lightning}}$

Immediate.

□

Corollary 2. If $t \in \mathcal{O}_1[\![\varphi]\!]$ then $t \in \mathcal{R}[\![\varphi]\!]$.

Proof. Induction on the derivation of $t \in \mathcal{O}_1[\![\varphi]\!]$, using the previous lemma. □

Lemma 13. If $t \in \mathcal{O}[\![\varphi]\!]$ then $t \in \mathcal{O}_1[\![\varphi]\!]^\dagger$

Proof. Induction on the derivation of $t \in \mathcal{O}[\![\varphi]\!]$, with nested induction on steps $\kappa \rightarrow^* \kappa'$. □

Corollary 3. If $t \in \mathcal{O}[\![\varphi]\!]$ then $t \in [\![\varphi]\!]$

Proof. Applying the previous lemma and Lem 11, we have:

$$\mathcal{O}[\![\varphi]\!] \subseteq \mathcal{O}_1[\![\varphi]\!]^\dagger \subseteq \mathcal{R}[\![\varphi]\!]^\dagger = [\![\varphi]\!]$$

□

3.3 Denotational traces are operational traces

Lemma 14. $\mathcal{O}_1[\![\varphi]\!]\,;\mathcal{O}_1[\![\psi]\!] \subseteq \mathcal{O}_1[\![\varphi;\psi]\!]$.

Proof. We prove that if $t \in \mathcal{O}_1[\![\varphi]\!]$ and $u \in \mathcal{O}_1[\![\psi]\!]$ then $t;u \in \mathcal{O}_1[\![\varphi;\psi]\!]$ by a straightforward induction on the derivation of $t \in \mathcal{O}_1[\![\varphi]\!]$. □

Lemma 15. $\mathcal{O}_1[\![\varphi]\!] \parallel \mathcal{O}_1[\![\psi]\!] \subseteq \mathcal{O}_1[\![\varphi \parallel \psi]\!]$.

Proof. We prove that if $t \in \mathcal{O}_1[\![\varphi]\!]$ and $u \in \mathcal{O}_1[\![\psi]\!]$ and $s \in t \parallel u$ then $s \in \mathcal{O}_1[\![\varphi;\psi]\!]$ by a straightforward induction on the derivation of $s \in t \parallel u$. □

Syntactic environment

$$
\begin{array}{rcl}
\eta & ::= & \emptyset \;\mid\; \eta, X \mapsto \varphi \;\mid\; \eta, f \mapsto F \;\mid\; \eta, x \mapsto v \\
\widehat{\emptyset} & \triangleq & \emptyset \\
\widehat{\eta, X \mapsto \varphi} & \triangleq & \widehat{\eta}, X \mapsto \mathcal{O}_1[\![\varphi]\!] \\
\widehat{\eta, f \mapsto F} & \triangleq & \widehat{\eta}, f \mapsto (v \mapsto \mathcal{O}_1[\![F\ v]\!]) \\
\widehat{\eta, x \mapsto v} & \triangleq & \widehat{\eta}, x \mapsto v
\end{array}
$$

Lemma 16. $\mathcal{R}[\![\varphi]\!]^{\widehat{\eta}} \subseteq \mathcal{O}_1[\![\eta\varphi]\!]$.

Proof. By induction on φ.

Case: $\boxed{\varphi;\psi}$

By induction, Lemma 14.

Case: $\boxed{\varphi \parallel \psi}$

By induction, Lemma 15.

Case: $\boxed{\varphi \vee \psi}$

By induction.

Case: $\boxed{\exists x.\varphi}$

By induction.

Case: $\boxed{\mu X.\varphi}$

By induction, fixpoint theorem.

Case: $\boxed{X}$

Immediate.

Case: $\boxed{\textsf{let } f = F \textsf{ in } \psi}$

By induction.

Case: $\boxed{F\ e}$

By induction (through procedures).

Case: $\boxed{\langle \forall \overline{x} : p,\ q \rangle}$

Immediate.

Case: $\boxed{\{p\}}$

Immediate.

□

Lemma 17. If $t \in \mathcal{O}_1[\![\varphi]\!]$ then $t \in \mathcal{O}[\![\varphi]\!]^\dagger$

Proof. Easy induction on the derivation of $t \in \mathcal{O}_1[\![\varphi]\!]$. □

Corollary 4. If φ closed then $[\![\varphi]\!] \subseteq \mathcal{O}[\![\varphi]\!]^\dagger$.

Proof. Appealing to the previous two lemmas and Lemmas 1 and 11 we have

$$[\![\varphi]\!] = (\mathcal{R}[\![\varphi]\!])^\dagger \subseteq \mathcal{O}_1[\![\varphi]\!]^\dagger \subseteq \mathcal{O}[\![\varphi]\!]^\dagger$$

□

3.4 Adequacy theorem

The proof of the adequacy theorem is given in the paper itself. It relies on the following result, which is a corollary to the work of the previous two subsections.

Corollary 5. If φ is closed then $[\![\varphi]\!] = \mathcal{O}[\![\varphi]\!]^\dagger$.

4 Laws of refinement

We prove the laws of refinement by proving the corresponding denotational refinement. Most of the time, we prove the denotational refinement using the raw semantics. Since † is monotonic, this will imply the refinement in the quotiented semantics. We also drop ρ in the proofs involving actions, where it is not relevant (since it is simply applied as a syntactic substitution).

Lemma 18 (DstL)**.** $(\varphi_1 \vee \varphi_2); \psi \equiv \varphi_1; \psi \vee \varphi_2; \psi$

Proof. Immediate by Lemma 7 (in particular, ; is continuous). □

Lemma 19 (DstR)**.** $\psi; (\varphi_1 \vee \varphi_2) \equiv \psi; \varphi_1 \vee \psi; \varphi_2$

Proof. Immediate by Lemma 7 (in particular, ; is continuous). □

Lemma 20 (Str_1)**.** $\exists x.\varphi; \psi \equiv \varphi; (\exists x.\psi)$

Proof. Immediate by Lemma 7 (in particular, ; is continuous). □

Lemma 21 (STR$_2$). $\exists x.\, \langle \forall \overline{y} : p,\ q\rangle \sqsubseteq \langle \forall \overline{y} : p,\ \exists x.q\rangle$

Proof.

Let $(\sigma, o) \in \mathrm{act}(p[v/x], q[v/x])$	
$(\sigma, o) \in \mathrm{act}(p, q[v/x])$	$x \notin p$
$\sigma = \sigma_1 \uplus \sigma_2,\ \ \sigma_1, \rho \models p \implies o = \sigma_1' \uplus \sigma_2,\ \ \sigma_1', \rho \models q[v/x]$	definition
$\sigma = \sigma_1 \uplus \sigma_2,\ \ \sigma_1, \rho \models p \implies o = \sigma_1' \uplus \sigma_2,\ \ \sigma_1', \rho \models \exists x.q$	$\therefore$
$(\sigma, o) \in \mathrm{act}(p, \exists x.q)$	definition

□

Lemma 22 (FRM). $\langle \forall \overline{x} : p,\ q\rangle \sqsubseteq \langle \forall \overline{x} : p * r,\ q * r\rangle$

Proof.

Let $(\sigma, o) \in \mathrm{act}(p, q)$	
If $\sigma, \rho \not\models p * r * \mathsf{true}$, done	†-closure
Otherwise, $\sigma = \sigma_0 \uplus \sigma_1 \uplus \sigma_2, \quad \sigma_0, \rho \models p, \quad \sigma_1, \rho \models r$	$\therefore$
$o = \sigma_0' \uplus \sigma_1 \uplus \sigma_2$ where $\sigma_0', \rho \models q$	assumption
$\sigma_0' \uplus \sigma_1, \rho \models q * r$	$\therefore$

□

Lemma 23 (EXT). If p exact then $\langle \forall \overline{x} : p,\ p\rangle \equiv \{\exists \overline{x}.p\}$

Proof. By faulting closure, and the definition of exact predicates. □

Lemma 24 (IDM$_1$). $\{p\}; \{p\} \equiv \{p\}$

Proof. Left-to-right: stuttering closure for non-faulting case, immediate for faulting case. Right-to-left: mumbling closure for non-faulting case, immediate for faulting case. □

Lemma 25 (IDM$_2$). $\{\exists \overline{x}.p\}; \langle \forall \overline{x} : p,\ q\rangle \equiv \langle \forall \overline{x} : p,\ q\rangle$

Proof. Left-to-right: stuttering closure for non-faulting case, immediate for faulting case. Right-to-left: mumbling closure for non-faulting case, immediate for faulting case. □

Lemma 26 (ASM). If r pure then $\langle \forall \overline{x} : p,\ q \wedge r\rangle \equiv \langle \forall \overline{x} : p,\ q\rangle\,; [r]$

Proof. From left-to-right: stuttering closure. From right-to-left: mumbling closure. In either case the extra assumption merely filters traces based on ρ, which is invariant. □

Lemma 27 (CSQ$_1$). If $\forall \overline{x}.\, p \Rightarrow p'$ and $\forall \overline{x}.\, q' \Rightarrow q$ then $\langle \forall \overline{x} : p',\ q'\rangle \sqsubseteq \langle \forall \overline{x} : p,\ q\rangle$

Proof.

Let $(\sigma, o) \in \mathrm{act}(p', q')$	
$\sigma = \sigma_1 \uplus \sigma_2,\ \ \sigma_1, \rho \models p' \implies o = \sigma_1' \uplus \sigma_2,\ \ \sigma_1', \rho \models q'$	definition
$\sigma = \sigma_1 \uplus \sigma_2,\ \ \sigma_1, \rho \models po = \sigma_1' \uplus \sigma_2,\ \ \sigma_1', \rho \models q$	assumption
$(\sigma, o) \in \mathrm{act}(p, q)$	definition

□

Lemma 28 (CSQ$_2$). If $q \Rightarrow p$ then $\{p\} \sqsubseteq \{q\}$

Proof. By faulting closure. □

5 Laws of fenced refinement

Lemma 29 (Inv)**.** $I, \theta \vdash \{I\} \equiv \mathsf{skip}$

Proof. Immediate by definition of fenced projection and assertions. □

Lemma 30 (Lift)**.** If $\varphi \sqsubseteq \psi$ then $I, \theta \vdash \varphi \sqsubseteq \psi$.

Proof. This law is a statement of the monotonicity of fenced projection, which holds because fenced projection is defined pointwise. □

Lemma 31 (SeqL)**.** $I, \theta \vdash \langle \forall \overline{x} : p,\ q \rangle ; \langle \forall \overline{x} : q * p',\ r \rangle \sqsubseteq \{I * \exists \overline{x}.p\}; \langle \forall \overline{x} : p * p',\ r \rangle$

Proof. Let

$$(\sigma_1, (\sigma, o); (\sigma', o'), \sigma_2) \in [\![I, \theta \vdash \langle \forall \overline{x} : p,\ q \rangle ; \langle \forall \overline{x} : q * p',\ r \rangle]\!]^\rho$$

If $\sigma_1 \uplus \sigma, \rho \not\models I * \exists \overline{x}.p$ we are done, by faulting closure. Otherwise, it must be that $\sigma_1, \rho \models \mathsf{true} * \exists \overline{x}.p$. By the semantics of actions it follows that $o = \sigma$, and the rest follows from stuttering closure. □

Lemma 32 (Stab)**.** If $\theta \sqsubseteq \langle \forall \overline{x} : q,\ q \rangle$ then $I, \theta \vdash \langle \forall \overline{x} : p,\ q \rangle ; \{\exists \overline{x}.q\} \sqsubseteq \langle \forall \overline{x} : p,\ q \rangle$

Proof. By definition of fenced projection and rely, easy to see that the assertion never fails. The rest follows by stuttering closure. □

6 Data$_2$

The soundness of the DATA$_2$ rule is proved by introducing a notion of *simulation* on traces. Simulation is used to generalize fenced refinement to an aribtrary number of concurrent object instances. To do this, it specifically relies on relating a concrete and abstract trace: the abstract trace provides the necessary information about which concurrent object instances are allocated at each point. Simulation is relative to the representation predicate r, the rely θ, an environment ρ, an abstract resource name α, and two subheaps: σ_c, which contains the concrete object instances, and σ_p, which contains private data allocated within method bodies. Within the definition of simulation, three additional subheaps appear: the unlabeled subheap σ, which described resources unrelated to the data abstraction being performed; σ_f, which is the frame of private resources allocated by other threads; and σ_a, which contains the abstract resources corresponding to σ_c. Notice that σ_a only shows up in the abstract traces, while σ_c, σ_p and σ_f show up only in the concrete traces.

The relevant definitions are given in Figure 2.

Lemma 33. We have

$$r, \theta, \rho, \alpha \models \mathrm{act}((\rho r)[\ell, y], (\rho r)[\ell, e] \wedge p) \cap [\![\theta[\ell]]\!]^\rho \leq \mathrm{act}(\ell \overset{\alpha}{\mapsto} y, \ell \overset{\alpha}{\mapsto} (\rho e) \wedge \rho p)$$

Lemma 34. If $r, \theta, \rho, \alpha \models T \leq U$ then $r, \theta, \rho, \alpha \models T^\dagger \leq U^\dagger$.

Trace simulation $r, \theta, \rho, \alpha, \sigma_c, \sigma_p \models t \leq u$

$$\textsc{TSEmpty}\ \frac{}{r, \theta, \rho, \alpha, \sigma_c, \sigma_p \models \epsilon \leq \epsilon} \qquad \textsc{TSWrong}\ \frac{\sigma_c \uplus \sigma_p \not\subseteq \sigma}{r, \theta, \rho, \alpha, \sigma_c, \sigma_p \models (\sigma, o); t \leq u}$$

$$\textsc{TSFault}\ \frac{r, \rho, \alpha \models \sigma_c \leq \sigma_a}{r, \theta, \rho, \alpha, \sigma_c, \sigma_p \models (\sigma \uplus \sigma_c \uplus \sigma_p \uplus \sigma_f, o); t \leq (\sigma \uplus \sigma_a, \lightning)}$$

$$\textsc{TSStep}\ \frac{\begin{array}{c} r, \theta, \rho, \alpha \models (\sigma_c, \sigma_a) \rightarrow^* (\sigma_c', \sigma_a') \\ \forall \sigma_c'', \sigma_a''.\ r, \theta, \rho, \alpha \models (\sigma_c', \sigma_a') \rightarrow^* (\sigma_c'', \sigma_a'') \implies r, \theta, \rho, \alpha, \sigma_c'', \sigma_p' \models t \leq u \end{array}}{r, \theta, \rho, \alpha, \sigma_c, \sigma_p \models (\sigma \uplus \sigma_c \uplus \sigma_p \uplus \sigma_f, \sigma' \uplus \sigma_c' \uplus \sigma_p' \uplus \sigma_f); t \leq (\sigma \uplus \sigma_a, \sigma' \uplus \sigma_a'); u}$$

Heap abstraction $r, \rho, \alpha \models \sigma_c \leq \sigma_a$

$$r, \rho, \alpha \models \sigma_c \leq \sigma_a \ \triangleq\ \begin{cases} \sigma_a = [\ell_1 \mapsto \alpha(v_1)] \cdots [\ell_n \mapsto \alpha(v_n)] \text{ and} \\ \sigma_c, \rho \models r[\ell_1, v_1] * \cdots * r[\ell_n, v_n] \end{cases}$$

Rely steps $r, \theta, \rho, \alpha \models (\sigma_c, \sigma_a) \rightarrow (\sigma_c', \sigma_a')$

$$r, \theta, \rho, \alpha \models (\sigma_c, \sigma_a) \rightarrow (\sigma_c', \sigma_a') \ \triangleq\ \begin{cases} \sigma_c = \sigma_0 \uplus \sigma_{cf} & \sigma_a = [\ell \mapsto \alpha(v)] \uplus \sigma_{af} \\ \sigma_c' = \sigma_0' \uplus \sigma_{cf} & \sigma_a' = [\ell \mapsto \alpha(v')] \uplus \sigma_{af} \\ \sigma_0, \rho \models r[\ell, v] & r, \rho, \alpha \models \sigma_{cf} \leq \sigma_{af} \\ \sigma_0', \rho \models r[\ell, v'] & (\sigma_0, \sigma_0') \in [\![\theta[\ell]]\!]^\rho \\ \vee & \\ r, \rho, \alpha \models \sigma_c \leq \sigma_a, \\ \sigma_c' = \sigma_c \uplus \sigma, \\ \sigma_a' = \sigma_a \uplus [\ell \mapsto \alpha(v)], \\ r, \rho, \alpha \models \sigma_c' \leq \sigma_a' \end{cases}$$

Simulation $r, \theta, \rho, \alpha \models T \leq U$

$$r, \theta, \rho, \alpha \models T \leq U \ \triangleq\ \forall t \in T. \forall \sigma_c. \exists u \in U.\ r, \theta, \rho, \alpha, \sigma_c, \emptyset \models t \leq u$$

Figure 2: Simulation

Lemma 35. If $r[\ell, -], \theta \models \varphi_1 \sqsubseteq \varphi_2$ and $r, \theta, \rho, \alpha \models [\![\varphi_2]\!]^\rho \leq [\![\varphi_3]\!]^\rho$ then $r, \theta, \rho, \alpha \models [\![\varphi_1]\!]^\rho \leq [\![\varphi_3]\!]^\rho$.

Lemma 36. If $r, \theta, \rho, \alpha \models T_1 \leq U_1$ and $r, \theta, \rho, \alpha \models T_2 \leq U_2$ then $r, \theta, \rho, \alpha \models T_1 \parallel T_2 \leq U_1 \parallel U_2$.

Proof. We prove this result pointwise, by proving the following: given σ_c, σ_a, σ_{p_1} and σ_{p_2} if, for all σ_c' such that $r, \theta, \rho, \alpha \models (\sigma_c, \sigma_a) \rightarrow^* (\sigma_c', \sigma_a')$ we have

- $r, \theta, \rho, \alpha, \sigma_c, \sigma_{p_1} \models t_1 \leq u_1$
- $r, \theta, \rho, \alpha, \sigma_c, \sigma_{p_2} \models t_2 \leq u_2$
- $\sigma_{p_1} \# \sigma_{p_2}$
- $t \in t_1 \parallel t_2$

then there exists a $u \in u_1 \parallel u_2$ such that

$$r, \theta, \rho, \alpha, \sigma_c, \sigma_{p_1} \uplus \sigma_{p_2} \models t \leq u$$

We prove by induction on the derivation of $t \in t_1 \parallel t_2$.

Case: $\boxed{\dfrac{t \in t_1 \parallel t_2}{t \in t_2 \parallel t_1}}$

By induction we have the result for $u \in u_2 \parallel u_1$, and so by applying the same symmetry rule we have it for $u \in u_1 \parallel u_2$.

Case: $\boxed{\dfrac{t \in t_1 \parallel t_2}{(\sigma, \sigma')t \in (\sigma, \sigma')t_1 \parallel t_2}}$

We proceed by case analysis on the last rule used in the derivation of

$$r, \theta, \rho, \alpha, \sigma_c, \sigma_{p_1} \models (\sigma, \sigma')t_1 \leq u_1$$

Subcase: $\boxed{\text{TSWrong}\ \dfrac{\sigma_c \uplus \sigma_{p_1} \not\subseteq \sigma}{r, \theta, \rho, \alpha, \sigma_c, \sigma_{p_1} \models (\sigma, o); t_1 \leq u_1}}$

Let $u \in u_1 \parallel u_2$ (always nonempty)
Apply TSWrong with t.

Subcase: $\boxed{\text{TSFault}\ \dfrac{r, \rho, \alpha \models \sigma_c \leq \sigma_a \qquad \sigma = \sigma_0 \uplus \sigma_c \uplus \sigma_{p_1} \uplus \sigma_f}{r, \theta, \rho, \alpha, \sigma_c, \sigma_{p_1} \models (\sigma, \sigma'); t_1 \leq (\sigma_0 \uplus \sigma_a, \lightning)}}$

Have $(\sigma_0 \uplus \sigma_a, \lightning) \in u_1 \parallel u_2$
Apply TSFault with t.

Subcase:

$$\frac{\begin{array}{l}\sigma = \sigma_0 \uplus \sigma_c \uplus \sigma_{p_1} \uplus \sigma_f \\ \sigma' = \sigma'_0 \uplus \sigma'_c \uplus \sigma'_{p_1} \uplus \sigma_f \\ r, \theta, \rho, \alpha \models (\sigma_c, \sigma_a) \to^* (\sigma'_c, \sigma'_a) \\ \forall \sigma''_c, \sigma''_a.\ r, \theta, \rho, \alpha \models (\sigma'_c, \sigma'_a) \to^* (\sigma''_c, \sigma''_a) \\ \qquad \Longrightarrow\ r, \theta, \rho, \alpha, \sigma''_c, \sigma'_{p_1} \models t \leq u\end{array}}{r, \theta, \rho, \alpha, \sigma_c, \sigma_p \models (\sigma, \sigma'); t \leq (\sigma_0 \uplus \sigma_a, \sigma'_0 \uplus \sigma'_a); u}\ \textsc{TSStep}$$

If $\sigma_{p_2} \not\leq \sigma_f$ apply TSWrong
Otherwise we can apply induction and TSStep

Case: $\boxed{(\sigma, \sigma')t \in (\sigma, \sigma') \parallel t_2}$

Similar to the TSStep case above.

Case: $\boxed{(\sigma, \lightning) \in (\sigma, \lightning) \parallel t_2}$

Similar to the TSFault case above.

□

Lemma 37. If $r, \theta, \rho, \alpha \models T_1 \leq U_1$ and $r, \theta, \rho, \alpha \models T_2 \leq U_2$ then $r, \theta, \rho, \alpha \models T_1; T_2 \leq U_1; U_2$.

Proof. Similar to the proof for parallel composition, but here we proceed by induction on the simulation judgment for $t_1 \in T_1$. □

Lemma 38. If $r, \theta, \rho, \alpha \models T_i \leq U_i$ then $r, \theta, \rho, \alpha \models \bigcup T_i \leq \bigcup U_i$.

Proof. Immediate: simulation is defined pointwise. □

Lemma 39. If α does not appear free in p or q then $r, \theta, \rho, \alpha \models \mathrm{act}(\rho p, \rho q) \leq \mathrm{act}(\rho p, \rho q)$.

Proof. Follows from locality (Lemma 1). □

Lemma 40. Suppose

- $r[\ell, -]$ precise
- p_i pure
- $r[\ell, -],\ \bigvee \overline{\theta_i} \vdash \varphi_i \sqsubseteq \theta_i$
- $\theta_i \sqsubseteq \langle \forall y : r[\ell, y],\ r[\ell, e_i] \wedge p_i \rangle$

Then

$$\begin{array}{rcl} r, \bigvee \overline{\theta_i}, \rho, \alpha & \models & \left[\!\!\left[\begin{array}{l} \textsf{let } f(\ell) = \langle \mathsf{emp},\ r[\ell, e] \rangle \textsf{ in} \\ \textsf{let } \overline{g_i(\ell, x) = \varphi_i} \textsf{ in } \psi \end{array} \right]\!\!\right]^\rho \\ & \leq & \left[\!\!\left[\begin{array}{l} \textsf{let } f(\ell) = \left\langle \mathsf{emp},\ \ell \overset{\alpha}{\mapsto} e \right\rangle \textsf{ in} \\ \textsf{let } \overline{g_i(\ell, x) = \left\langle \forall y : \ell \overset{\alpha}{\mapsto} y,\ \ell \overset{\alpha}{\mapsto} e_i \wedge p_i \right\rangle} \textsf{ in } \psi \end{array} \right]\!\!\right]^\rho \end{array}$$

Proof. By induction on ψ, using the preceeding lemmas to handle each case. □

Lemma 41. If $r, \theta, \rho, \alpha, \emptyset, \emptyset \models t \leq u$ then either

- $t = s \uplus t_c$, $u = s \uplus_\alpha u_\alpha$ with t_c, u_α sequential and $\mathrm{fst}(t_c) = \sigma_c \uplus \sigma_p$ and $\mathrm{fst}(u_\alpha) = \sigma_a$ with $r, \theta, \rho, \alpha \models \sigma_c \leq \sigma_a$ or
- $t = (s \uplus t_c); t'$, $u = (s \uplus_\alpha u_\alpha)u'$ with t_c, u_α sequential and $\mathrm{fst}(t_c) = \sigma_c \uplus \sigma_p$ and $\mathrm{fst}(u_\alpha) = \sigma_a$ with $r, \theta, \rho, \alpha \models \sigma_c \leq \sigma_a$ and $\mathrm{lst}(t_c) \not\subseteq \mathrm{fst}(t')$.

Proof. Induction on the derivation of $r, \theta, \rho, \alpha, \sigma_c, \sigma_p \models t \leq u$. □

Corollary 6. DATA$_2$ is sound.

Proof. By Lemmas 40 and 41. □