A Proposal for Weak-Memory Local Reasoning

Ian Wehrman (UT Austin)
Josh Berdine (Microsoft Research)

Overview

Project: a separation logic with x86-TSO semantics.

Goal: nice proofs of concurrent data structures.

Racy by design; can't avoid the memory model.

Small, ubiquitous, hard to get right, even worse to debug.

Overview

Project: a separation logic with x86-TSO semantics.

Goal: nice proofs of concurrent data structures.

Racy by design; can't avoid the memory model.

Small, ubiquitous, hard to get right, even worse to debug.

Caveat: work is in progress.

Lots of details remain; no soundness proof yet.

x86-TSO

Store: $[e] \coloneqq f_p$

enqueue (e,f) on p^{th} write buffer, if e is allocated in heap.

x86-TSO

Store: $[e] \coloneqq f_p$

enqueue (e,f) on p^{th} write buffer, if e is allocated in heap.

Load: $x \coloneqq [e]_p$

x=v if (e,v) is the most recent write to e in the p^{th} buffer

else if heap(e) = v.

x86-TSO

Store: $[e] \coloneqq f_p$

enqueue (e,f) on p^{th} write buffer, if e is allocated in heap.

Load: $x \coloneqq [e]_p$

x=v if (e,v) is the most recent write to e in the p^{th} buffer

else if heap(e) = v.

Fence: fence_p

commit all writes from the p^{th} buffer to mem in FIFO order.

Local reasoning

Restrict reasoning to **relevant resources**.

(strong) partial heaps.

Local reasoning

Restrict reasoning to **relevant resources**.

(strong) partial heaps.

(weak) partial heaps **&** partial write buffers.

Basic resources

Represented with predicates:

(strong)

emp – empty heap.

points to

$e \mapsto f$ – exactly one heap cell at address e with value f.

Basic resources

Represented with predicates:

(strong)

emp – empty heap.

points to → $e \mapsto f$ – exactly one heap cell at address e with value f.

(weak)

emp – empty heap **&** empty write buffers.

leads to → $e \rightharpoonup_p f$ – empty heap **&** one write to e on p^{th} buffer with value f,

or heap cell at address e with value f **&** empty write buffers.

Barriers

Represented with logical operations $\mathbf{bar}_p$ that flush writes pending on p^{th} buffer.

$\mathbf{bar}_p(e \to_p f)$ – heap cell at e with value f & empty write buffers.

Barriers

Represented with logical operations $\mathbf{bar}_p$ that flush writes pending on p^{th} buffer.

$\mathbf{bar}_p(e \rightarrow_p f)$ – heap cell at e with value f & empty write buffers.

$e \mapsto f \triangleq \mathbf{bar}_p(e \rightarrow_p f)$

Structured resources

Represented with separating conjunctions:

(strong)

$P * Q$ – disjoint union of heaps.

Structured resources

Represented with separating conjunctions:

(strong)

$P * Q$ – disjoint union of heaps.

(weak)

interleaving → $P * Q$ – disjoint union of heaps & **interleaved** write buffers.

sequential → $P ; Q$ – disjoint union of heaps & **concatenated** write buffers.

Examples

$e \rightsquigarrow_p e' * f \rightsquigarrow_p f'$ – 2 writes to distinct locations in either order; either may have not yet committed.

Examples

$e \rightsquigarrow_p e' * f \rightsquigarrow_p f'$ – 2 writes to distinct locations in either order; either may have not yet committed.

$e \rightsquigarrow_p e' \,;\, f \rightsquigarrow_p f'$ – 2 writes to distinct locations in order; if f has committed then so has e.

Examples

$e \rightsquigarrow_p e' * f \rightsquigarrow_p f'$ – 2 writes to distinct locations in either order; either may have not yet committed.

$e \rightsquigarrow_p e' ; f \rightsquigarrow_p f'$ – 2 writes to distinct locations in order; if f has committed then so has e.

$e \mapsto e' ; f \rightsquigarrow_p f'$ – 2 writes to distinct locations in order; e in heap; f may have not yet committed.

Examples

$e \rightsquigarrow_p e' * f \rightsquigarrow_p f'$ – 2 writes to distinct locations in either order; either may have not yet committed.

$e \rightsquigarrow_p e' ; f \rightsquigarrow_p f'$ – 2 writes to distinct locations in order; if f has committed then so has e.

$e \mapsto e' ; f \rightsquigarrow_p f'$ – 2 writes to distinct locations in order; e in heap; f may have not yet committed.

$e \mapsto f ; e \rightsquigarrow_p f'$ – inconsistent.

Counting permissions

Weaken sequential conjunction to describe successive writes to a location.

Counting permissions

Weaken sequential conjunction to describe successive writes to a location.

e.g., $(x \rightarrow_{p,-1} 1) \,;\, (x \rightarrow_{p,-1} 2) \,;\, (x \rightarrow_{p,2} 3)$

- $n \geq 0$ indicates the most recent write…
  - … and no more than n previous writes.
- $n < 0$ indicates a previous write.

Counting permissions

Weaken sequential conjunction to describe successive writes to a location.

e.g., $(x \rightharpoondown_{p,-1} 1) ; (x \rightharpoondown_{p,-1} 2) ; (x \rightharpoondown_{p,2} 3)$

$n \geq 0$ indicates the most recent write…

… and no more than n previous writes.

$n < 0$ indicates a previous write.

e.g., $(x \rightharpoondown_{p,0} 1) ; (x \rightharpoondown_{p,-1} 2) \models$ false

Program logic

Sep. logic for C-like language w/atomic regions.

- Resource invariants describe shared memory, ala CSL.
- Semantics of atomic commands does **not** include fences.

Program logic

Sep. logic for C-like language w/atomic regions.

Resource invariants describe shared memory, ala CSL.

Semantics of atomic commands does **not** include fences.

$R \vdash \{ P \}\ c\ \{ Q \}$ means:

executions of c that start in an $R * P$ state

1) always respect invariant R,

2) have no memory errors, and

3) terminate in an $R * Q$ state, or else diverge.

Invariant expansion

Invariants are heap-and-variable assertions.

e.g., $R \triangleq \exists ab\ .\ x \mapsto_0 a * y \mapsto_0 b * (a{=}1 \vee b{=}1)$.

Invariant expansion

Invariants are heap-and-variable assertions.

e.g., $R \triangleq \exists ab \,.\, x \mapsto_0 a * y \mapsto_0 b * (a=1 \vee b=1)$.

exp(R) denotes states that always preserve invariant R while committing writes.

e.g., $(x \mapsto_{-1} 0 * y \mapsto_{-1} 1)\ ,\ x \rightharpoonup_{p,1} 1\ ,\ y \rightharpoonup_{p,1} 0 \vDash \mathbf{exp}(R)$.

Frame rules

int. frame

$$\frac{R \vdash \{\ P\ \}\ c\ \{\ Q\ \} \quad \mathrm{mod}(c) \cap \mathrm{fv}(F) = \varnothing}{R \vdash \{\ F * P\ \}\ c\ \{\ F * Q\ \}}$$

Frame rules

int. frame

$$\frac{R \vdash \{\, P \,\}\ c\ \{\, Q \,\} \qquad \mathrm{mod}(c) \cap \mathrm{fv}(F) = \varnothing}{R \vdash \{\, F * P \,\}\ c\ \{\, F * Q \,\}}$$

seq. frame

$$\frac{R \vdash \{\, P \,\}\ c\ \{\, Q \,\} \qquad \mathrm{mod}(c) \cap \mathrm{fv}(F) = \varnothing}{R \vdash \{\, F \,;\, P \,\}\ c\ \{\, F \,;\, Q \,\}}$$

Axioms

load

$$R \vdash \{\ e \rightsquigarrow_{p,n} f\ ,\ Q\ \}\ x \coloneqq [e]_p\ \{\ (e \rightsquigarrow_{p,n} f\ ,\ Q) \wedge x = f\ \}$$

where $n \geq 0$ and $x \notin fv(e,f,R)$.

Axioms

load

$R \vdash \{\ e \rightarrow_{p,n} f\ ,\ Q\ \}\ x \coloneqq [e]_p\ \{\ (e \rightarrow_{p,n} f\ ,\ Q) \wedge x = f\ \}$

where $n \geq 0$ and $x \notin fv(e,f,R)$.

store

$R \vdash \{\ e \rightarrow_{p,n} e'\ ,\ Q\ \}\ [e] \coloneqq f_p\ \{\ e \rightarrow_{p,-1} e'\ ,\ Q\ ,\ e \rightarrow_{p,n+1} f\ \}$

where $n \geq 0$.

Axioms

load

$$R \vdash \{\ e \rightsquigarrow_{p,n} f\ ,\ Q\ \}\ x \coloneqq [e]_p\ \{\ (e \rightsquigarrow_{p,n} f\ ,\ Q) \wedge x = f\ \}$$

where $n \geq 0$ and $x \notin fv(e,f,R)$.

store

$$R \vdash \{\ e \rightsquigarrow_{p,n} e'\ ,\ Q\ \}\ [e] \coloneqq f_p\ \{\ e \rightsquigarrow_{p,-1} e'\ ,\ Q\ ,\ e \rightsquigarrow_{p,n+1} f\ \}$$

where $n \geq 0$.

fence

$$R \vdash \{\ Q\ \}\ \mathrm{fence}_p\ \{\ \mathbf{bar}_p(Q)\ \}$$

Memory management

allocate

$R \vdash \{\ \mathbf{emp}\ \}\ x \coloneqq \mathrm{new}(e)_p\ \{\ x \rightarrow_{p,0} e\ \}$

Memory management

allocate

$$R \vdash \{\ \mathbf{emp}\ \}\ x := \mathrm{new}(e)_p\ \{\ x \rightarrow_{p,0} e\ \}$$

dispose

$$R \vdash \{\ e \rightarrow_{p,0} -\ \}\ \mathrm{free}(e)_p\ \{\ \mathbf{emp}\ \}$$

Concurrency rules

parallel

$$\frac{R \vdash \{\ P\ \}\ c\ \{\ Q\ \} \quad R \vdash \{\ P'\ \}\ c'\ \{\ Q'\ \} \quad (\S)}{R \vdash \{\ P * P'\ \}\ c \parallel c' \{\ Q * Q'\ \}}$$

Concurrency rules

parallel

$$\frac{R \vdash \{\ P\ \}\ c\ \{\ Q\ \} \quad R \vdash \{\ P'\ \}\ c'\ \{\ Q'\ \} \quad (\S)}{R \vdash \{\ P * P'\ \}\ c \parallel c' \{\ Q * Q'\ \}}$$

atomic

$$\frac{\mathbf{emp} \vdash \{\ R * P\ \}\ c\ \{\ \mathbf{exp}(R) * Q\ \} \quad (\dagger)}{R \vdash \{\ P\ \}\ \langle\!\langle c \rangle\!\rangle\ \{\ Q\ \}}$$

Concurrency rules

parallel

$$\frac{R \vdash \{\, P \,\}\; c \;\{\, Q \,\} \qquad R \vdash \{\, P' \,\}\; c' \;\{\, Q' \,\} \quad (\S)}{R \vdash \{\, P * P' \,\}\; c \parallel c' \{\, Q * Q' \,\}}$$

atomic

$$\frac{\mathbf{emp} \vdash \{\, R * P \,\}\; c \;\{\, \mathbf{exp}(R) * Q \,\} \quad (\dagger)}{R \vdash \{\, P \,\}\; \langle\!\langle c \rangle\!\rangle \;\{\, Q \,\}}$$

share

$$\frac{R \vdash \{\, P \,\}\; c \;\{\, Q \,\}}{\mathbf{emp} \vdash \{\, \mathbf{exp}(R) * P \,\}\; c \;\{\, \mathbf{exp}(R) * Q \,\}}$$

Very simple example

$$\frac{}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\} \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel\!| \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle \{\, y{=}4 \,\}}$$

Very simple example

store ax.

$$\frac{}{\mathbf{emp} \vdash \{\ x \mapsto_0 4\ \}\ [x] \coloneqq 4_p\ \{\ x \mapsto_{-1} 4\ ;\ x \rightharpoonup_{p,1} 4\ \}}$$

$$\frac{}{\mathbf{emp} \vdash \{\ x \mapsto_0 4\ \}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel\!| \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\ y{=}4\ \}}$$

Very simple example

store ax.

conseq.

$$\dfrac{\dfrac{}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, x \mapsto_{-1} 4 \ ;\ x \rightharpoonup_{p,1} 4 \,\}}}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, \mathbf{exp}(x \mapsto_0 4) \,\}}$$

$$\dfrac{}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, y{=}4 \,\}}$$

Very simple example

store ax.

$$\dfrac{}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, x \mapsto_{-1} 4\ ;\ x \rightharpoonup_{p,1} 4 \,\}}$$

conseq.

$$\dfrac{}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, \mathbf{exp}(x \mapsto_0 4) \,\}}$$

atomic

$$\dfrac{}{x \mapsto_0 4 \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle\ \{\, \mathbf{emp} \,\}} \qquad \dfrac{}{\ldots \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, y=4 \,\}}$$

$$\dfrac{}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, y=4 \,\}}$$

Very simple example

store ax.

$$\frac{}{\mathbf{emp} \vdash \{\ x \mapsto_0 4\ \}\ [x] \coloneqq 4_p\ \{\ x \mapsto_{-1} 4\ ;\ x \rightharpoonup_{p,1} 4\ \}}$$

conseq.

$$\frac{\mathbf{emp} \vdash \{\ x \mapsto_0 4\ \}\ [x] \coloneqq 4_p\ \{\ x \mapsto_{-1} 4\ ;\ x \rightharpoonup_{p,1} 4\ \}}{\mathbf{emp} \vdash \{\ x \mapsto_0 4\ \}\ [x] \coloneqq 4_p\ \{\ \mathbf{exp}(x \mapsto_0 4)\ \}}$$

atomic

$$\frac{\mathbf{emp} \vdash \{\ x \mapsto_0 4\ \}\ [x] \coloneqq 4_p\ \{\ \mathbf{exp}(x \mapsto_0 4)\ \}}{x \mapsto_0 4 \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle\ \{\ \mathbf{emp}\ \}} \qquad \frac{}{\ldots \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\ y{=}4\ \}}$$

parallel

$$\frac{x \mapsto_0 4 \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle\ \{\ \mathbf{emp}\ \} \qquad \ldots \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\ y{=}4\ \}}{x \mapsto_0 4 \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\ \mathbf{emp} * y{=}4\ \}}$$

$$\frac{}{\mathbf{emp} \vdash \{\ x \mapsto_0 4\ \}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\ y{=}4\ \}}$$

Very simple example

store ax.

$$\dfrac{}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, x \mapsto_{-1} 4 \,;\, x \rightharpoonup_{p,1} 4 \,\}}$$

conseq.

$$\dfrac{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, x \mapsto_{-1} 4 \,;\, x \rightharpoonup_{p,1} 4 \,\}}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, \mathbf{exp}(x \mapsto_0 4) \,\}}$$

atomic

$$\dfrac{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ [x] \coloneqq 4_p\ \{\, \mathbf{exp}(x \mapsto_0 4) \,\}}{x \mapsto_0 4 \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle\ \{\, \mathbf{emp} \,\}} \qquad \dfrac{}{\ldots \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, y{=}4 \,\}}$$

parallel

$$\dfrac{x \mapsto_0 4 \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle\ \{\, \mathbf{emp} \,\} \qquad \ldots \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, y{=}4 \,\}}{x \mapsto_0 4 \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, \mathbf{emp} * y{=}4 \,\}}$$

share

$$\dfrac{x \mapsto_0 4 \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, \mathbf{emp} * y{=}4 \,\}}{\mathbf{emp} \vdash \{\, \mathbf{exp}(x \mapsto_0 4) \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, \mathbf{exp}(x \mapsto_0 4) * y{=}4 \,\}}$$

conseq.

$$\dfrac{\mathbf{emp} \vdash \{\, \mathbf{exp}(x \mapsto_0 4) \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, \mathbf{exp}(x \mapsto_0 4) * y{=}4 \,\}}{\mathbf{emp} \vdash \{\, x \mapsto_0 4 \,\}\ \langle\!\langle [x] \coloneqq 4_p \rangle\!\rangle \parallel \langle\!\langle y \coloneqq [x]_q \rangle\!\rangle\ \{\, y{=}4 \,\}}$$

Intermission

Heap-only invariants yield a simple but weak logic.

- Can't reason about the *history* of writes to shared state.
- Each write must preserve the invariant.

Intermission

Heap-only invariants yield a simple but weak logic.

- Can't reason about the *history* of writes to shared state.
- Each write must preserve the invariant.

Split assertions into shared *heap* & local *buffer* parts.

- Heap part: summarization of writes checked against invariant.
- Buffer part: exact history of local writes to shared state.
- … really nice, come talk to us!

Stubborn example

Let $c_w \triangleq \langle\!\langle [d] \coloneqq 1_p \rangle\!\rangle \,;\, \langle\!\langle [r] \coloneqq 1_p \rangle\!\rangle$

Stubborn example

Let $c_w \triangleq \langle\!\langle [d] \coloneqq 1_p \rangle\!\rangle \,;\, \langle\!\langle [r] \coloneqq 1_p \rangle\!\rangle$

Let $c_r \triangleq \langle\!\langle x \coloneqq [r]_q \rangle\!\rangle \,;\, \langle\!\langle y \coloneqq [d]_q \rangle\!\rangle$

Stubborn example

Let $c_w \triangleq \langle\!\langle [d] := 1_p \rangle\!\rangle \,;\, \langle\!\langle [r] := 1_p \rangle\!\rangle$

Let $c_r \triangleq \langle\!\langle x := [r]_q \rangle\!\rangle \,;\, \langle\!\langle y := [d]_q \rangle\!\rangle$

Let $R \triangleq \exists d', r' \,.\, d \mapsto_0 d' * r \mapsto_0 r' * (r'=1 \Rightarrow d'=1)$

Spec: $\mathbf{emp} \vdash \{ R \}\ c_w \parallel c_r\ \{ x=1 \Rightarrow y=1 \}$

Stubborn example

Let $c_w \triangleq \langle\!\langle [d] := 1_p \rangle\!\rangle \,;\, \langle\!\langle [r] := 1_p \rangle\!\rangle$

Let $c_r \triangleq \langle\!\langle x := [r]_q \rangle\!\rangle \,;\, \langle\!\langle y := [d]_q \rangle\!\rangle$

Let $R \triangleq \exists d', r' .\ d \mapsto_0 d' * r \mapsto_0 r' * (r'=1 \Rightarrow d'=1)$

Spec: $\mathbf{emp} \vdash \{ R \}\ c_w \parallel c_r\ \{ x=1 \Rightarrow y=1 \}$

True, but can't prove that c_w preserves invariant.

Stubborn proof (1)

$$\frac{}{R \vdash \{\, \mathbf{emp} \,\} \langle\!\langle [d] \coloneqq 1 \rangle\!\rangle \{\, \mathbf{emp} \,\}}$$

Stubborn proof (1)

store

$$\frac{}{\mathbf{emp} \vdash \{\ d \mapsto d'\ \}\ [d] \coloneqq 1\ \{\ d \mapsto d'\ ;\ d \rightarrow 1\ \}}$$

$$\frac{}{R \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [d] \coloneqq 1 \rangle\!\rangle\ \{\ \mathbf{emp}\ \}}$$

Stubborn proof (1)

$$\dfrac{\dfrac{}{\mathbf{emp} \vdash \{\ d \mapsto d'\ \}\ [d] \coloneqq 1\ \{\ d \mapsto d'\ ;\ d \rightarrow 1\ \}}\ \text{store}}{\mathbf{emp} \vdash \{\ d \mapsto d' * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1)\ \}\ [d] \coloneqq 1\ \{\ (d \mapsto d'\ ;\ d \rightarrow 1) * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1)\ \}}\ \text{frame}$$

$$\dfrac{}{R \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [d] \coloneqq 1 \rangle\!\rangle\ \{\ \mathbf{emp}\ \}}$$

Stubborn proof (1)

store

frame

conseq.

$$\dfrac{\dfrac{\dfrac{}{\mathbf{emp} \vdash \{\ d \mapsto d'\ \}\ [d] \coloneqq 1\ \{\ d \mapsto d'\ ;\ d \rightsquigarrow 1\ \}}}{\mathbf{emp} \vdash \{\ d \mapsto d' * r \mapsto r' * (r'=1 \Rightarrow d'=1)\ \}\ [d] \coloneqq 1\ \{\ (d \mapsto d'\ ;\ d \rightsquigarrow 1) * r \mapsto r' * (r'=1 \Rightarrow d'=1)\ \}}}{\mathbf{emp} \vdash \{\ d \mapsto d' * r \mapsto r' * (r'=1 \Rightarrow d'=1)\ \}\ [d] \coloneqq 1\ \{\ (d \mapsto d' * r \mapsto r')\ ;\ d \rightsquigarrow 1 * (r'=1 \Rightarrow d'=1)\ \}}$$

$$\dfrac{}{R \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [d] \coloneqq 1 \rangle\!\rangle\ \{\ \mathbf{emp}\ \}}$$

Stubborn proof (1)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{emp} \vdash \{\, d \mapsto d' \,\}\ [d] := 1\ \{\, d \mapsto d' \,;\, d \to 1 \,\}}\ \text{store}}{\mathbf{emp} \vdash \{\, d \mapsto d' * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [d] := 1\ \{\, (d \mapsto d' \,;\, d \to 1) * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1) \,\}}\ \text{frame}}{\mathbf{emp} \vdash \{\, d \mapsto d' * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [d] := 1\ \{\, (d \mapsto d' * r \mapsto r') \,;\, d \to 1 * (r'{=}1 \Rightarrow d'{=}1) \,\}}\ \text{conseq.}}{\mathbf{emp} \vdash \{\, d \mapsto d' * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [d] := 1\ \{\, \mathbf{exp}(R) \,\}}\ \text{conseq.}$$

$$\dfrac{}{R \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [d] := 1 \rangle\!\rangle\ \{\, \mathbf{emp} \,\}}$$

OK: $(d \mapsto d' * r \mapsto r') \,;\, d \to 1 * (r'{=}1 \Rightarrow d'{=}1) \quad \Rightarrow \quad \mathbf{exp}(R)$

Stubborn proof (1)

$$\begin{array}{lc}
\text{store} & \\ \cline{2-2}
 & \mathbf{emp} \vdash \{ d \mapsto d' \}\ [d] := 1\ \{ d \mapsto d' ; d \rightarrow 1 \} \\
\text{frame} & \\ \cline{2-2}
 & \mathbf{emp} \vdash \{ d \mapsto d' * r \mapsto r' * (r'=1 \Rightarrow d'=1) \}\ [d] := 1\ \{ (d \mapsto d' ; d \rightarrow 1) * r \mapsto r' * (r'=1 \Rightarrow d'=1) \} \\
\text{conseq.} & \\ \cline{2-2}
 & \mathbf{emp} \vdash \{ d \mapsto d' * r \mapsto r' * (r'=1 \Rightarrow d'=1) \}\ [d] := 1\ \{ (d \mapsto d' * r \mapsto r') ; d \rightarrow 1 * (r'=1 \Rightarrow d'=1) \} \\
\text{conseq.} & \\ \cline{2-2}
 & \mathbf{emp} \vdash \{ d \mapsto d' * r \mapsto r' * (r'=1 \Rightarrow d'=1) \}\ [d] := 1\ \{ \mathbf{exp}(R) \} \\
\text{exists} & \\ \cline{2-2}
 & \mathbf{emp} \vdash \{ R \}\ [d] := 1\ \{ \mathbf{exp}(R) \} \\
\text{atomic} & \\ \cline{2-2}
 & R \vdash \{ \mathbf{emp} \}\ \langle\!\langle [d] := 1 \rangle\!\rangle\ \{ \mathbf{emp} \}
\end{array}$$

OK: $(d \mapsto d' * r \mapsto r') ; d \rightarrow 1 * (r'=1 \Rightarrow d'=1) \quad \Rightarrow \quad \mathbf{exp}(R)$

Stubborn proof (2?)

$$\frac{}{\mathrm{R} \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [\mathrm{r}] \coloneqq 1 \rangle\!\rangle\ \{\ \mathbf{emp}\ \}}$$

Stubborn proof (2?)

store

$$\frac{}{\mathbf{emp} \vdash \{\ r \mapsto r'\ \}\ [r] \coloneqq 1\ \{\ r \mapsto r'\ ;\ r \rightarrow 1\ \}}$$

$$\frac{}{R \vdash \{\ \mathbf{emp}\ \}\ \langle\!\langle [r] \coloneqq 1 \rangle\!\rangle\ \{\ \mathbf{emp}\ \}}$$

Stubborn proof (2?)

store

frame

$$\frac{}{\textbf{emp} \vdash \{\, r \mapsto r' \,\}\ [r] \coloneqq 1\ \{\, r \mapsto r' \,;\, r \rightarrow 1 \,\}}$$

$$\textbf{emp} \vdash \{\, r \mapsto r' * d \mapsto d' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [r] \coloneqq 1\ \{\, (r \mapsto r' \,;\, r \rightarrow 1) * d \mapsto d' * (r'{=}1 \Rightarrow d'{=}1) \,\}$$

$$\frac{}{R \vdash \{\, \textbf{emp} \,\}\ \langle\!\langle [r] \coloneqq 1 \rangle\!\rangle\ \{\, \textbf{emp} \,\}}$$

Stubborn proof (2?)

store

frame

conseq.

$$\cfrac{\cfrac{\cfrac{}{\mathbf{emp} \vdash \{\, r \mapsto r' \,\}\ [r] \coloneqq 1\ \{\, r \mapsto r' \,;\, r \rightarrow 1 \,\}}}{\mathbf{emp} \vdash \{\, r \mapsto r' * d \mapsto d' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [r] \coloneqq 1\ \{\, (r \mapsto r' \,;\, r \rightarrow 1) * d \mapsto d' * (r'{=}1 \Rightarrow d'{=}1) \,\}}}{\mathbf{emp} \vdash \{\, d \mapsto d' * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [r] \coloneqq 1\ \{\, (d \mapsto d' * r \mapsto r') \,;\, r \rightarrow 1 * (r'{=}1 \Rightarrow d'{=}1) \,\}}$$

$$\cfrac{}{R \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [r] \coloneqq 1 \rangle\!\rangle\ \{\, \mathbf{emp} \,\}}$$

Stubborn proof (2?)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{emp} \vdash \{\, r \mapsto r' \,\}\ [r] \coloneqq 1\ \{\, r \mapsto r' \,;\, r \rightsquigarrow 1 \,\}}\ \text{store}}{\mathbf{emp} \vdash \{\, r \mapsto r' * d \mapsto d' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [r] \coloneqq 1\ \{\, (r \mapsto r' \,;\, r \rightsquigarrow 1) * d \mapsto d' * (r'{=}1 \Rightarrow d'{=}1) \,\}}\ \text{frame}}{\mathbf{emp} \vdash \{\, d \mapsto d' * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [r] \coloneqq 1\ \{\, (d \mapsto d' * r \mapsto r') \,;\, r \rightsquigarrow 1 * (r'{=}1 \Rightarrow d'{=}1) \,\}}\ \text{conseq.}}{\mathbf{emp} \vdash \{\, d \mapsto d' * r \mapsto r' * (r'{=}1 \Rightarrow d'{=}1) \,\}\ [r] \coloneqq 1\ \{\, \mathbf{exp}(R) \,\}}\ \text{conseq.}$$

$$\vdots$$

$$\dfrac{}{R \vdash \{\, \mathbf{emp} \,\}\ \langle\!\langle [r] \coloneqq 1 \rangle\!\rangle\ \{\, \mathbf{emp} \,\}}$$

NOT OK: $(d \mapsto d' * r \mapsto r') \,;\, r \rightsquigarrow 1 * (r'{=}1 \Rightarrow d'{=}1) \quad \Rightarrow \quad \mathbf{exp}(R)$

Post-mortem

Can't prove $R \vdash \{\, \textbf{emp} \,\}\ [r] \coloneqq 1\ \{\, \textbf{emp} \,\}$ …

… because first write was absorbed into the invariant.

Want to prove instead something like:

$$R \vdash \{\, \ldots ; d \rightsquigarrow 1 \,\}\ [r] \coloneqq 1\ \{\, \ldots ; d \rightsquigarrow 1 ; r \rightsquigarrow 1 \,\}$$

Idea: heap state is shared, but writes are *local*.

Splitting permissions

Weaken ($*$) to allow read-only heap sharing:

$$x \mapsto_{0,½} 4 * x \mapsto_{0,½} 4 \equiv x \mapsto_{0,½} 4$$

Splitting permissions

Weaken ($*$) to allow read-only heap sharing:

$$x \mapsto_{0,½} 4 * x \mapsto_{0,½} 4 \;\equiv\; x \mapsto_{0,½} 4$$

$$x \mapsto_{0,1} 4 * x \mapsto_{-,-} - \;\equiv\; \text{false}$$

Splitting permissions

Weaken ($*$) to allow read-only heap sharing:

$x \mapsto_{0,½} 4 * x \mapsto_{0,½} 4 \;\equiv\; x \mapsto_{0,½} 4$

$x \mapsto_{0,1} 4 * x \mapsto_{-,-} - \;\equiv\; \text{false}$

$x \mapsto_{-1,1} 3 \,;\, x \rightsquigarrow_{p,1,1} 4 \;\equiv\; (x \mapsto_{-1,1} 3 \vee x \mapsto_{-1,1} 4) * (x \mapsto_{-1,1} 3 \,;\, x \rightsquigarrow_{p,1,1} 4)$

Splitting permissions

Weaken ($*$) to allow read-only heap sharing:

$x \mapsto_{0,½} 4 * x \mapsto_{0,½} 4 \;\equiv\; x \mapsto_{0,½} 4$

$x \mapsto_{0,1} 4 * x \mapsto_{-,-} - \;\equiv\; \text{false}$

$x \mapsto_{-1,1} 3 \,;\, x \rightsquigarrow_{p,1,1} 4 \;\equiv\; (x \mapsto_{-1,1} 3 \vee x \mapsto_{-1,1} 4) * (x \mapsto_{-1,1} 3 \,;\, x \rightsquigarrow_{p,1,1} 4)$

store, allocate & dispose axioms: full splitting permission.

Stubborn example, again

Let $R_{c,s} \triangleq \exists d', r' \,.\, d \mapsto_{c,s} d' * r \mapsto_{c,s} r' * (r'=1 \Rightarrow d'=1)$

Important equivalences:

$R_{0,1} \equiv R_{0,½} * R_{0,½}$ and

$R_{-1,1} \,;\, d \rightsquigarrow_{1,1} 1 \,;\, r \rightsquigarrow_{1,1} 1 \equiv (R_{0,½}) * (R_{-1,½} \,;\, d \rightsquigarrow_{1,½} 1 \,;\, r \rightsquigarrow_{1,½})$

Strong specs now provable:

$R_{0,½} \vdash \{ R_{0,½} \}\ c_w\ \{ R_{-1,½} \,;\, d \rightsquigarrow_{1,½} 1 \,;\, r \rightsquigarrow_{1,½} \}$ and

$\mathbf{emp} \vdash \{ R_{0,1} \}\ c_w \parallel c_r\ \{ x=1 \Rightarrow y=1 \}$