A new view of Concurrent Separation Logic with Abstract State

December 8, 2015

1 Concurrent rules

1.1 Notation

- **Locks:** $l\boxdot\xrightarrow[v]{\pi} R$, "$l$ is a lock with share π, partial view v, and invariant R". The view v must be a CSPCMC as described bellow in 1.3 and R must be a function of it.
- **Holds:** $\boxed{\text{🔒}}\, l\ v$, "Holds l with full view v". It is not necessary to specify the invariant R of the lock l. This differs from Hobor's notation, because his *release* rule, doesn't require a lock share.
- **Thread:** $t\bigcirc\!\!\rightarrow Q$, "thread t has postcondition Q". It is loosely define by the following equation, but should be treated opaquely:

$$l\bigcirc\!\!\rightarrow Q \triangleq l\boxdot\xrightarrow{\CIRCLE} S \qquad \text{where } S = (Q * l\boxdot\xrightarrow{\LEFTcircle} S)$$

1.2 Rules for threads and locks

The views, can be though of as a repository. On acquire, a thread *pulls* the sum of all views $v_t = v_o \oplus v_s$, which represents the reality $R(v_t)$. This old value, is recorded in the hold, $\boxed{\text{🔒}}\, l\ v_t$, which also ensures ownership of the lock. On release, the thread *commits* the *diff* v_δ to it's local repo, changing $v_s \to v_s \oplus v_\delta$ and *pushes* the changes by releasing $R(v_t \oplus v_\delta)$ and $\boxed{\text{🔒}}\, l\ v_t$. the rest is similar to Hobor's thesis.

$$\text{makelock}\ \frac{R \text{ positive}^? \qquad R \text{ precise}^?}{\{l \mapsto 0\}\ \text{makelock}\ l\ \{l\boxdot\xrightarrow[v]{\bullet} R * \boxed{\text{🔒}}\, l\ v\}}\ [1]$$

$$\text{freelock}\ \frac{}{\{l\boxdot\xrightarrow[\cdot]{\bullet} R * \boxed{\text{🔒}}\, l\ \_\}\ \text{freelock}\ l\ \{l \mapsto 0\}}$$

[1] There is some work pending on the restrictions for rule *makelock*. Since R is now a function, it is not obvious what positive and precise mean. It is not even obvious we need those!

$$\text{split}\ \frac{\pi_1 \oplus \pi_2 = \pi \quad v_1 \oplus v_2 = v}{l \boxdot \xrightarrow[v_1]{\pi_1} R * l \boxdot \xrightarrow[v_2]{\pi_2} R \longleftrightarrow l \boxdot \xrightarrow[v]{\pi} R}$$

$$\text{acq}\ \frac{}{\{l \boxdot \xrightarrow[v_s]{\pi} R\}\ \text{acquire}\ l\ \{\exists v_o, \text{🔓} l\ (v_s \oplus v_o) * R(v_s \oplus v_o) * l \boxdot \xrightarrow[v_s]{\pi} R\}}$$

$$\text{rel}\ \frac{\exists v_{s'}.\ v_s \oplus v_\delta = v_{s'}}{\{\text{🔓} l\ v_t * R(v_t \oplus v_\delta) * l \boxdot \xrightarrow[v_s]{\pi} R\}\ \text{release}\ l\ \{l \boxdot \xrightarrow[v_s \oplus v_\delta]{\pi} R\}}$$

$$\text{spwn}\ \frac{l\ \text{fresh in}\ F}{\{(f : \{P\vec{y}\}\{Q\}) \wedge (Px)[\vec{b}/\vec{y}] * F\}\ \text{Spawn f}(\vec{b})\ \{l \bigcirc\!\!\rightarrow (Qx) * F\}}$$

$$\text{join}\ \frac{}{\{l \bigcirc\!\!\rightarrow Q\}\ \text{Join}\ l\ \{Q\}}$$

$$\text{gather}\ \frac{l \boxdot \xrightarrow[v_s]{\bullet} R * \text{🔓} l\ v_t}{v_s = v_t}$$

The release and gather rules will often be used in this weaker, but more intuitive form:

$$\text{rel}^*\ \frac{}{\{\text{🔓} l\ (v_s \oplus v_o) * R(v_s \oplus v_o \oplus v_\delta) * l \boxdot \xrightarrow[v_s]{\pi} R\}\ \text{release}\ l\ \{l \boxdot \xrightarrow[v_s \oplus v_\delta]{\pi} R\}}$$

$$\text{gather}^*\ \frac{l \boxdot \xrightarrow[v_s]{\bullet} R * \text{🔓} l\ (v_s \oplus v_o)}{v_s = v_s \oplus v_o}$$

1.3 Cross-split Partial Commutative Monoid with Cores (CSPCMC).

First, we need a good name for this structure, they are not even a real a subset of PCMs (no unique unit)...

A CSPCMC (or whatever we call it) is a triple $(P, \oplus, \widehat{\ })$ of a set P, a partial binary operation $\oplus : P \times P \nrightarrow P$, and a function $\widehat{\ } : P \to P$ satisfying the following rules.

- Commutativity and associativity:

$$\text{comm}\ \frac{p_1 \oplus p_2 = p}{p_2 \oplus p_1 = p} \qquad \text{assoc}\ \frac{p_1 \oplus p_2 = p_{12} \qquad p_{12} \oplus p_3 = p_{123}}{\exists p_{23}, p_2 \oplus p_3 = p_{23} \wedge p_1 \oplus p_{23} = p_{123}}$$

- Well behaved cores:

$$\text{unit}\ \frac{}{p \oplus \widehat{p} = p} \qquad \text{compat}\ \frac{p_1 \oplus p_2 = p}{\widehat{p_1} = \widehat{p}}$$

$$\text{dup}\frac{p \oplus p = p}{p = \widehat{p}} \qquad\qquad \text{split}\frac{p_1 \oplus p_2 = \widehat{p}}{p_1 \oplus p_1 = p_1}$$

- Cross-split property (this one is debatable, but I include it for now):

$$\text{cross-split}\frac{a \oplus b = p \qquad c \oplus d = p}{\exists ac\ ad\ bc\ bd.\ \ ac \oplus ad = a \quad bc \oplus bd = b \quad ac \oplus bc = c \quad ad \oplus bd = d}$$

First notice there is no unique unit, which allows structures like resources (which have multiple, but compatible according to compt, units). Similarly, there is no cancellativity. This remains to be proven but seems like it's not needed. There is no full argument for or against cross-split, but it seems that is necessary.