Reduction from RA to SC using fences

Ori Lahav        Viktor Vafeiadis

31 August 2017
For TSO, it suffices to have a fence between every racy write & subsequent racy read.

For RA, we need more fences. Recall the IRIW example:

**Independent reads of independent writes (IRIW)**

\[
\begin{align*}
    x := 1 & \\
    a := x; & // 1 \\
    b := y & // 0 \\
    c := y; & // 1 \\
    d := x & // 0 \\
    y := 1
\end{align*}
\]
What is the semantics of SC fences?

From C11, we had:

\[ \text{eco} \triangleq (rf \cup mo \cup rb)^+ \quad \text{(extended coherence order)} \]

\[ \text{psc}_F \triangleq [F^{sc}]; (hb \cup hb; \text{eco}; hb); [F^{sc}] \quad \text{(partial SC fence order)} \]

and required that \( \text{psc}_F \) is acyclic.

That is,

**Definition (RA consistency with fences)**

An execution graph \( G \) is **RA-consistent** iff there exists some modification order \( mo \) for \( G \) such that:

- \( G \) is complete,
- \( (po \cup rf)^+|_{loc} \cup mo \cup rb \) is acyclic, and
- \( \text{psc}_F \) is acyclic.
Alternative definition of RA consistency

Theorem

An execution graph $G$ is RA-consistent iff there exists a total order $\text{sc}$ on $G. F^{\text{sc}}$ and a modification order $\text{mo}$ for $G$ such that:

- $G$ is complete,
- $(\text{po} \cup \text{rf} \cup \text{sc})^+$ is irreflexive, and
- $(\text{po} \cup \text{rf} \cup \text{sc})^*; \text{eco}$ is irreflexive.
Theorem

Let $G$ be an RA-consistent execution graph. If

- For every $G$-racy events $a, b$, if $\langle a, b \rangle \in (G.p_0 \cup G.rf)^+$,
  then $\langle a, c \rangle, \langle c, b \rangle \in (G.p_0 \cup G.rf)^+$ for some fence event $c$.

Then, $G$ is SC-consistent.
Recall SC-consistency: $\text{po} \cup \text{rf} \cup \text{mo} \cup \text{rb}$ is acyclic.

Let $\text{hb} \triangleq (\text{po} \cup \text{rf} \cup \text{sc})^+$ and $\text{K} \triangleq \text{eco} \setminus \text{hb}$.

It suffices to prove: $\text{hb} \cup \text{K}$ is acyclic.

Consider minimal cycle in $(\text{hb} \cup \text{K})$.

- Cycles with $\leq 1$ $\text{K}$-edges disallowed by RA consistency.
- Cycle with two $\text{K}$-edges:
Proof of the simple reduction theorem (1/2)

Recall:

- Recall SC-consistency: $\text{po} \cup \text{rf} \cup \text{mo} \cup \text{rb}$ is acyclic.
- Let $\text{hb} \triangleq (\text{po} \cup \text{rf} \cup \text{sc})^+$ and $\text{K} \triangleq \text{eco} \setminus \text{hb}$.
- It suffices to prove: $\text{hb} \cup \text{K}$ is acyclic.

Consider minimal cycle in $(\text{hb} \cup \text{K})$.

- Cycles with $\leq 1$ $\text{K}$-edges disallowed by RA consistency.
- Cycle with two $\text{K}$-edges:

\[
\begin{array}{c}
\text{a} \\
\text{hb}
\end{array} \quad \begin{array}{c} K \\
\text{f}_1 \quad \text{f}_2 \quad \text{hb}
\end{array} \quad \begin{array}{c}
\text{b} \\
\text{d}
\end{array}
\]

\[
\begin{array}{c}
\text{c} \\
\text{hb}
\end{array} \quad \begin{array}{c} K \\
\text{d}
\end{array}
\]

\[
\begin{array}{c}
\text{f}_1 \\
\text{f}_2
\end{array}
\]

\[
\begin{array}{c}
\text{a} \\
\text{b}
\end{array}
\]

\[
\begin{array}{c} K \\
\text{c}
\end{array}
\]
Proof of the simple reduction theorem (1/2)

Recall:
- Recall SC-consistency: \( \text{po} \cup \text{rf} \cup \text{mo} \cup \text{rb} \) is acyclic.
- Let \( \text{hb} \overset{\triangle}{=} (\text{po} \cup \text{rf} \cup \text{sc})^+ \) and \( K \overset{\triangle}{=} \text{eco} \setminus \text{hb} \).
- It suffices to prove: \( \text{hb} \cup K \) is acyclic.

Consider minimal cycle in \( (\text{hb} \cup K) \).
- Cycles with \( \leq 1 \) \( K \)-edges disallowed by RA consistency.
- Cycle with two \( K \)-edges:
Finally, consider a cycle with three or more $K$-edges.
Finally, consider a cycle with three or more $K$-edges.

\begin{figure}
\centering
\begin{tikzpicture}
  \node (a) at (0,0) {a};
  \node (b) at (2,0) {b};
  \node (c) at (2,2) {c};
  \node (d) at (0,2) {d};
  \node (f1) at (1,1) {$f_1$};
  \node (f2) at (1,3) {$f_2$};

  \draw[red,->] (a) to (b);
  \draw[blue,->] (a) to (d);
  \draw[blue,->] (b) to (c);
  \draw[blue,->] (c) to (d);
  \draw[red,->] (d) to (a);
  \draw[blue,->] (b) to (f1);
  \draw[blue,->] (f1) to (f2);
  \draw[blue,->] (f2) to (c);
  \draw[red,->] (f2) to (b);

  \node at (1.5, 1.1) {$K$};
  \node at (0.5, 1.1) {$K$};
  \node at (0.5, 0.2) {$hb$};
  \node at (1.5, 0.2) {$hb$};
  \node at (0.5, 1.5) {$hb$};
  \node at (1.5, 1.5) {$hb$};
  \node at (0.5, 2.1) {$(hb \cup K)^+$};
\end{tikzpicture}
\caption{Diagram of the cycle with $K$-edges.}
\end{figure}
Finally, consider a cycle with three or more $K$-edges.

\[
\begin{array}{c}
\text{a} \quad \text{b} \\
\downarrow & \downarrow \\
\text{f}_1 & \text{f}_2 \\
\downarrow & \downarrow \\
\text{d} \quad \text{c} \\
\end{array}
\]
More advanced reduction theorem

**Theorem**

Let $G$ be a **WW-race-free RA-consistent** execution, and there exists a set $B \subseteq G.E$ of **protected events** such that:

1. $hb \triangleq (G.po \cup G.rf)^+$ is total on $B$.
2. If $a$ races with $b$ in $G$, then either $a \in B$ or $b \in B$.
3. For every $G$-racy write/update event $a \in B$ and $G$-racy read event $b \in B$, if $\langle a, b \rangle \in hb$, then $\langle a, c \rangle, \langle c, b \rangle \in hb$ for some fence event $c$.
4. For every $G$-racy write/update event $a \notin B$ and $G$-racy read event $b \notin B$, if $\langle a, b \rangle \in hb$, then $\langle a, c \rangle, \langle c, b \rangle \in hb$ for some fence or protected event $c$.

Then, $G$ is **SC-consistent**.
Proof sketch

Note
Because of WW-race-freedom, if \( \langle a, b \rangle \in K \), then a is a read and b is a write (or update).

Consider minimal cycle in \((hb \cup K)\).

- Cycle with two \(K\)-edges:
Proof sketch

Note

Because of WW-race-freedom, if \( \langle a, b \rangle \in K \), then \( a \) is a read and \( b \) is a write (or update).

Consider minimal cycle in \( (hb \cup K) \).

- Cycle with two \( K \)-edges:

\[
\begin{align*}
\in B & \quad K \\
a:R & \quad K \\
\notin B & \\
b:W & \\
d:W & \quad K \\
c:R & \quad K \\
hb & \quad hb \\
\end{align*}
\]
Proof sketch

Note

Because of WW-race-freedom, if \( \langle a, b \rangle \in K \), then \( a \) is a read and \( b \) is a write (or update).

Consider minimal cycle in \((hb \cup K)\).

- Cycle with two \( K \)-edges:

\[
\begin{array}{c}
\in B \\
\in B \\
\notin B \\
\notin B \\
\notin B \\
\in B \\
\in B \\
\notin B
\end{array}
\]

\[
\begin{array}{c}
a:R \\
b:W \\
d:W \\
c:R \\
hb \\
hb \\
hb \\
hb
\end{array}
\]
Proof sketch

Because of WW-race-freedom, if \( \langle a, b \rangle \in K \), then \( a \) is a read and \( b \) is a write (or update).

Consider minimal cycle in \((hb \cup K)\).

- Cycle with two \( K \)-edges:
Proof sketch

Note

Because of WW-race-freedom, if \( \langle a, b \rangle \in K \), then \( a \) is a read and \( b \) is a write (or update).

Consider minimal cycle in \((hb \cup K)\).

- Cycle with two \( K \)-edges:

\[
\begin{align*}
\in B & \quad \in B \\
a:R & \quad b:W \\
\notin B & \quad \notin B \\
d:W & \quad c:R
\end{align*}
\]
Proof sketch

Note

Because of WW-race-freedom, if $\langle a, b \rangle \in K$, then $a$ is a read and $b$ is a write (or update).

Consider minimal cycle in $(hb \cup K)$.

- Cycle with two $K$-edges:

```
\begin{tikzpicture}[node distance=2.5cm, auto]
  \node (a) {a:R};
  \node (b) [right of=a] {b:W};
  \node (c) [below of=b] {c:R};
  \node (d) [below of=a] {d:W};
  \node (hb) [above of=a] {hb};
  \node (f1) [right of=hb] {f$_1$};

  \path[->, red] (a) edge node {$K$} (b);
  \path[->, blue] (hb) edge (a);
  \path[->, blue] (f1) edge (hb);
  \path[->, blue] (d) edge (c);
  \path[->, red] (d) edge node {$K$} (c);
  \path[->, red] (c) edge node {$\notin B$} (b);
  \path[->, blue] (b) edge (hb);
  \path[->, blue] (hb) edge (b);
  \path[->, blue] (a) edge (d);
  \path[->, blue] (d) edge (a);

\end{tikzpicture}
```
Proof sketch

Note
Because of WW-race-freedom, if $\langle a, b \rangle \in K$, then $a$ is a read and $b$ is a write (or update).

Consider minimal cycle in $(hb \cup K)$.

- Cycle with two $K$-edges:

```
        ⊆ B
       /   \\  \\
 a:R   →   K   → b:W
     /    \   /    \    /    \  \\  \\  \\
   hb    f₁    K    f₂    hb
       ↓   /    ↓    ↓   /    ↓   \\
 d:W ←   /    /    /    /    /   c:R
      ↓   ↓   ↓   ↓   ↓  \\
   ⊆ B  ⊆ B  ⊆ B
          \  /  \  /
         f₁ f₂ hb
```

Proof sketch

**Note**
Because of WW-race-freedom, if \( \langle a, b \rangle \in K \), then \( a \) is a read and \( b \) is a write (or update).

Consider minimal cycle in \((hb \cup K)\).

- Cycle with two \( K \)-edges:
Applying the theorem to RCU

rcu_quiescent_state():
    \texttt{rc[get\_my\_tid()] := gc; fence();}

rcu_thread_offline():
    \texttt{rc[get\_my\_tid()] := 0; fence();}

rcu_thread_online():
    \texttt{rc[get\_my\_tid()] := gc; fence();}

synchronize_rcu():
    \texttt{local was\_online := (rc[get\_my\_tid()] \neq 0);}  
    \texttt{if was\_online then rc[get\_my\_tid()] := 0; lock();}
    \texttt{gc := gc + 1; fence();}
    \texttt{for i := 1 to N do wait (rc[i] \in \{0,gc\});}
    \texttt{unlock();}
    \texttt{if was\_online then rc[get\_my\_tid()] := gc; fence();}