C/C++ Concurrency
Axioms

\[
\begin{align*}
  a &= X; \\
  \text{if}(a == 1) \quad &b &= Y; \\
  \quad &Y &= 1; \\
\end{align*}
\]

\[
\begin{align*}
  \text{if}(b == 1) \quad &a &= X; \\
  \quad &Y &= 1; \\
\end{align*}
\]

(A) \( a = b = 1 \) is not possible.

\[
\begin{align*}
  \text{if}(b == 1) \quad &b &= Y; \\
  \quad &X &= 1; \\
\end{align*}
\]

(B) \( a = b = 1 \) is possible.

Two programs share an execution.

Causality cycle: \((\text{po} \cup \text{rf})\).
Axioms

Causality cycle: \((p_0 \cup r_f)\).

Allow causality cycle:
- Justifies outcome in (A) which is not possible.
- Violates data-race-freedom guarantee.
- Justifies (B) outcome.
- Allows read-write reordering.

Disallow causality cycle:
- Restrict undesired outcome in (A).
- Provides data-race-freedom guarantee.
- Restricts (B) outcome.
- Restrict read-write reordering in compiler and architectures.

Tradeoff: Data-race-freedom guarantee vs efficient compilation.
Compilation.

Outcome of execution:
\[ O(X) \triangleq \{(\text{e.Loc, e.wval}) \mid e \in X.E \cap WU. \{e\}; X.mo = \emptyset\} \]

Outcome of program P:
\[ O(P) = \{O(X) \mid X \in [[P]]\} \]

Given a compilation
\[ P_{C11} \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \cdots \Rightarrow P_{\text{arch}} \]

Correctness requires:
\[ O(P_{C11}) \supseteq O(P_1) \supseteq O(P_2) \supseteq \cdots \supseteq O(P_{\text{arch}}) \]
Exercise

Study examples from [POPL’15].