# C/C++ Concurrency

# Axioms

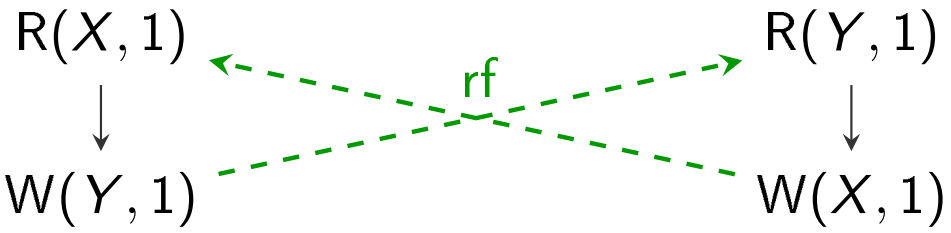
$$\begin{array}{l} a = X; \\ \text{if}(a == 1) \\ \qquad Y = 1; \end{array} \quad \Bigg\| \quad \begin{array}{l} b = Y; \\ \text{if}(b == 1) \\ \qquad X = 1; \end{array}$$

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

$$\begin{array}{l} a = X; \\ Y = 1; \end{array} \quad \Bigg\| \quad \begin{array}{l} b = Y; \\ \text{if}(b == 1) \\ \qquad X = 1; \end{array}$$

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

Two programs share an execution.




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

# Axioms

Causality cycle: $(\mathsf{po} \cup \mathsf{rf})$.

Allow causality cycle:

- Justifies outcome in (A) which is not possible.
- Violates data-race-freedom guarantee.
- Justitifes (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:

$$O(X) \triangleq \{(e.\mathsf{Loc}, e.\mathsf{wval}) \mid e \in X.\mathsf{E} \cap \mathsf{WU}.\ [\{e\}]; X.\mathsf{mo} = \emptyset\}$$

Outcome of peogram P:

$$O(P) = \{O(X) \mid X \in [[P]]\}$$

Given a compilation

$$P_{\mathsf{C11}} \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \cdots \Rightarrow P_{\mathsf{arch}}$$

Correctness requires:

$$O(P_{\mathsf{C11}}) \supseteq O(P_1) \supseteq O(P_2) \supseteq \cdots \supseteq O(P_{\mathsf{arch}})$$

# Exercise

Study examples from [POPL'15].