

The Happens-Before Relationship

Here is a short summary of the definitions involved in the Happens-Before relation.

1. A trace is a sequence of events.

```
Events E ::= start(T)
          | end(T)
          | read(T,x,v)
          | write(T,x,v)
          | lock(T,x)
          | unlock(T,x)
```

2. Let $E1 < E2$ be the ordering of events as they appear in the trace.

3. Define happens-before ordering $<:$ in a trace R as follows:

$E1 <: E2$ iff $E1 < E2$ and one of the following holds:

- a) $\text{thread}(E1) = \text{thread}(E2)$
- b) $E1$ is $\text{unlock}(T1,x)$ and $E2$ $\text{lock}(T2,x)$
- c) there exists $E3$ with $E1 <: E3$ and $E3 <: E2$

4. Updates are visible based on the following rules. For a trace r containing $EW == \text{write}(T1,x,v1)$ and $ER == \text{read}(T2,x,v2)$:

EW "is not visible" to ER if

- $ER <: EW$
- There exists some event $EW2 == \text{write}(T,x,v3)$ such that $EW <: EW2 <: R$

Otherwise EW is visible at ER

5. A data race takes place when there are two events in trace R that
 - access the same memory location
 - at least one is a write
 - they are unordered according to happens-before