Lecture 29 (Execution Witness)

1 Program Order (po) Edge

1. Edges between reads and writes (both directions)
2. Edges between instruction and synch (both directions)

2 Read From (rf) Edge

1. Write to read edge
2. Can be external and internal
3. If writes are atomic, then `rfe` edges are global
4. `rfi` is not global because of LSQ forwarding

3 Write Serialization (ws) Edge

1. Originates from PLSC and coherence
2. Is always global

small is beautiful - not sir talking about class size

4 From Read (fr) Edge

1. Read to write edge
2. It is global

5 Synchronization Edge (so) Edge

1. Exist for synchronization variables
2. All updates to such variables are synch

6 Summary

1. `ws`, `fr` and `so` are always global
2. `ppo` $\subset$ `po` be global
3. `grf` $\subset$ `rf` be global
4. `grf` $=$ `rf` and `ppo` $=$ `po` will hold in case of SC
5. `ghb` $=$ `ppo` $\cup$ `ws` $\cup$ `fr` $\cup$ `grf`
6. Subset is decided according to memory model

7 Cycles in EW

1. We perform a toposort to get the sequential execution
2. If sort is not possible, then we have a cycle

8 Access Graph

1. `up` edges between accesses to same location in same thread
2. `PLSC` $\equiv$ `up` $\cup$ `ws` $\cup$ `fr` $\cup$ `grf`

9 Data and Control Dependence

1. We add causal dependencies (`dep` edges) for if (and similar) statements
2. Otherwise we might have “thin air read”
3. `causal graph $\equiv$`rf`$\cup$`gpo`$\cup$`dep‘