

# HW1 – Memory Model Outcomes

CS 554, Fall 2025

Due Sep. 5

## 1 Glossary

- Sequential consistency (SC) is a memory model in which all operations execute according to a single total order (which is consistent with per-thread program order), all threads agree on this total order, and each read from a location reads the value of the most recent write to that location. Equivalently, SC is the model in which each memory operation takes full effect when it is performed and is immediately visible to all other threads.
- Total store order (TSO) is a memory model in which all *write* operations execute according to a single total order (which is consistent with per-thread program order). Each read operation may read any write that is not earlier in the store order than one that has already been read by that thread. Equivalently, TSO is the model in which each thread's writes are inserted into a buffer for that thread, and writes in the buffer may later propagate to other threads (but may not be reordered within the buffer).
- Partial store order (PSO) is a memory model in which all write operations *to the same location* execute according to a single total order (which is consistent with per-thread program order), but writes to different locations are unordered. Each read operation may read any write that is not earlier in its location's store order than one that has already been read by that thread. Equivalently, PSO is the model in which each thread has a write buffer for *each location*, and writes in all buffers may later propagate to other threads (but may not be reordered within each buffer).

## 2 Problems

This is a written homework with 7 problems. For the first 6 problems, say whether the program running under the given memory model can produce the listed outcome, and explain why or why not. Assume that all memory locations are initialized to 0.

1. Model: SC

Program:

$$\begin{array}{l} \text{store}(x, 1) \quad \parallel \quad \text{store}(y, 1) \\ \text{r1} = \text{load}(y) \quad \parallel \quad \text{r2} = \text{load}(x) \end{array}$$

Result:  $r1 = 1, r2 = 0$

2. Model: SC

Program:

$$\begin{array}{l} \text{store}(x, 1) \quad \parallel \quad \text{store}(y, 1) \\ \text{r1} = \text{load}(y) \quad \parallel \quad \text{r2} = \text{load}(x) \end{array}$$

Result:  $r1 = 0, r2 = 0$

3. Model: TSO

Program:

$$\begin{array}{l} \text{store}(x, 1) \quad \parallel \quad \text{store}(y, 1) \\ \text{r1} = \text{load}(y) \quad \parallel \quad \text{r2} = \text{load}(x) \end{array}$$

Result:  $r1 = 0, r2 = 0$

4. Model: PSO

Program:

```
store(x, 1) || store(y, 1)
r1 = load(y) || r2 = load(x)
```

Result: r1 = 0, r2 = 0

5. Model: TSO

Program:

```
store(x, 1) || r1 = load(y)
store(y, 2) || r2 = load(x)
```

Result: r1 = 2, r2 = 0

6. Model: PSO

Program:

```
store(x, 1) || r1 = load(y)
store(y, 2) || r2 = load(x)
```

Result: r1 = 2, r2 = 0

7. Explain why any outcome of a program that is possible under SC is also possible under TSO.