HW3 – Language Memory Models

CS 554, Fall 2025
Due October 27

1 Rules

1.1 Sequentially Consistent Atomics and Non-Atomics (SC+NA)

1. There is a total order to on all *atomic* operations, consistent with po.
2. There is a partial reads-from order rf that orders each write $W\ \ell\ v$ before any reads that read from it. The synchronizes-with order sw is exactly the rf order on atomic operations.
3. The happens-before order $\mathsf{hb} := \mathsf{po} \cup \mathsf{sw}$ is acyclic.
4. Each atomic read $R_{\text{at}}\ \ell\ v$ reads from the immediately preceding write to ℓ in to.
5. "No intervening writes": if $w_1 \xrightarrow{\mathsf{rf}} r$ and there is another write w_2 to the same location such that $w_1 <_{\mathsf{hb}} w_2$, then it is not the case that $w_2 <_{\mathsf{hb}} r$. Equivalently, if $w_1 \xrightarrow{\mathsf{rf}} r$ and there is another write w_2 to the same location ℓ such that $w_1 <_{\mathsf{hb}} w_2$, we can draw a from-read edge $r \xrightarrow{\mathsf{fr}_\ell} w_2$, and $\mathsf{hb} \cup \mathsf{fr}_\ell$ is acyclic. (Note that for atomic locations, the "no intervening writes" condition follows from the previous rule.)
6. If a and b are non-atomic operations on the same location, at least one of them is a write, and neither $a <_{\mathsf{hb}} b$ nor $b <_{\mathsf{hb}} a$, then a and b are a data race.

1.2 Release-Acquire Atomics and Non-Atomics (RA+NA)

1. There is a partial reads-from order rf that orders each write $W\ \ell\ v$ before any reads that read from it. The synchronizes-with order sw is exactly the rf order on atomic operations.
2. The happens-before order $\mathsf{hb} := \mathsf{po} \cup \mathsf{sw}$ is acyclic.
3. "No intervening writes": for each location ℓ, there is a total order mo_ℓ on all writes to ℓ that is consistent with po. If $w_1 \xrightarrow{\mathsf{rf}} r$ and there is another write w_2 such that $w_1 <_{\mathsf{mo}_\ell} w_2$, then it is not the case that $w_2 <'_{\mathsf{hb}} r$, where $\mathsf{hb}'_\ell := \mathsf{hb} \cup \mathsf{mo}_\ell$. Equivalently, if $w_1 \xrightarrow{\mathsf{rf}} r$ and there is another write w_2 to the same location ℓ such that $w_1 <_{\mathsf{mo}_\ell} w_2$, we can draw a from-read edge $r \xrightarrow{\mathsf{rf}_\ell} w_2$, and $\mathsf{hb} \cup \mathsf{mo}_\ell \cup \mathsf{fr}_\ell$ is acyclic.

4. If a and b are non-atomic operations on the same location, at least one of them is a write, and neither $a <_{\mathsf{hb}} b$ nor $b <_{\mathsf{hb}} a$, then a and b are a data race.

2 Problems

This is a written homework with 3 problems, each of which has multiple parts.

1. (a) Is SC+NA a stronger model or a weaker one than RA+NA?

   (b) Write a program that has an allowed behavior under one of the two models that is disallowed under the other model. Draw the execution graph for the behavior under each model, and show why the behavior is disallowed in the stronger model. (Hint: The differences will be clearest in a program where all memory operations are atomic.)

2. Consider the following C program:

```
*y = 1;                  ||  while(atomic_load(x) != 1);
atomic_store(x, 1);      ||  b = *y;
*y = 2;                  ||
```

   (a) Draw an execution of this program under RA+NA that has a data race. Be sure to indicate which two operations race with each other. You may check your answer using CppMem.

   (b) How could we change the program to remove this data race?

3. Suppose we were trying to write a highly optimizing compiler that performed optimizations on atomic operations, and allowed it to reorder atomic writes past non-atomic writes to other locations.

   (a) Intuitively, why would this mess up the behavior of programs?

   (b) Give an example of a program whose allowed behaviors would be changed by this optimization.

   (c) Is there any safe optimization a compiler could perform involving atomic operations? Why or why not? (This will be very generously graded—just write out your thoughts.)