

Assignment on Memory Models

1. Consider the SC model. For each program, enumerate the values that can be read at the commented line, and explain why:

(a)

```
Initially, x = y = 0.
x = 1; || x = 2; || while (y < 2);
y = x; || y = x; || r = x; // What values can be read?
```

(b)

```
Initially, x = y = 0.
x = 2; || b = y; || x = 1;
a = x; || if (b == 1) || c = x; // What values can be read?
if (a != 2) || x = 3; ||
y = 1; ||
```

2. Consider the happens-before memory model (HMM). For each program, tell whether the behavior in question is allowed, and explain why:

(a)

```
Initially, x = 0.
r1 = x; || r2 = x;
x = 1; || x = 2;
```

Behavior in question: r1 = 2, r2 = 1.

(b)

```
Initially, x = y = 0.
r1 = x; || r3 = y;
r2 = 1 + r1 * r1 - r1; || x = r3;
y = r2;
```

Behavior in question: r1 = r2 = 1.

(c)

```
Initially, x = y = z = 0.
r1 = x; || do {
if (r1 == 0) || r2 = y;
y = 1; || r3 = z;
else || } while (r2 + r3 == 0);
z = 1; || x = 1;
```

Behavior in question: r1 = r3 = 1, r2 = 0.

3. Consider the C/C++11 memory model. For each program, given the commented constraints, enumerate the values that can be read at the commented line, and explain why:

(a)

```
atomic_int x = 0;
x.store(1, relaxed); || x.load(relaxed); // this reads 1
                    || x.load(relaxed); // What values can be read?
```

(b)

```
atomic_int x = 0;
x.store(1, relaxed);
x.load(relaxed); // What values can be read?
x.store(2, relaxed);
```

(c)

```
atomic_int x = 0;
x.load(acquire); // this reads 2 || x.load(acquire); // What values can be read?
x.store(1, release);           || x.store(2, release);
```