The Theory & Practice of Concurrent Programming

Lectured by Azalea Raad and Alastair Donaldson

Typed by Aris Zhu Yi Qing

October 14, 2021

1 Synchronisation Paradigms

1.1 Properties in Asynchronous computation

1. Safety
   - Nothing bad happens ever
   - If it is violated, it is done by a finite computation
2. Liveness
   - Something good happens eventually
   - Cannot be violated by a finite computation

1.2 Problems in Asynchronous computation

1. Mutual Exclusion (Safety)
   - **cannot** be solved by transient communication or interrupts
   - **can** be solved by shared variables that can be read or written
2. No Deadlock (Liveness)

1.3 Protocols in Asynchronous computation

1. Flag Protocol:
   - Raise flag
   - While A's flag is up
     - Lower flag
     - Wait for A's flag to go down
     - Raise flag
   - Do something
   - Lower flag
2. Producer/Consumer:
   - For A(producer), while flag is up wait. So when flag becomes down, do something, then raise the flag.
   - For B(consumer), while flag is down, wait. So when flag becomes up, do something, then put down the flag.
3. Readers/Writers:
   - Each thread `i` has `size[i]` counter. Only it increments or decrements.
   - To get object's size, a thread reads a "snapshot" of all counters.
   - This eliminates the bottleneck of "having exclusive access to the common counter".

1.4 Performance Measurement

Amdahl's law:

$$\text{Speedup} = \frac{\text{1-thread execution time}}{n\text{-thread execution time}} = \frac{1}{1-p+\frac{p}{n}},$$

where p is the fraction of the algorithm having parallel execution, and n is the number of threads.