Lecture: Replication and Consistency
Exercise Sheet 4
`https://pl.cs.uni-kl.de/homepage/de/teaching/ws19/rac/`

1 Yet Another Consensus Protocol

Consider three threads, A, B, and C, each of which has a MRSW register, X_A, X_B, and X_C, that it alone can write and the others can read. In addition, each pair shares a `RMWRegister` register that provides only a `compareAndSet()` method: A and B share R_{AB}, B and C share R_{BC}, and A and C share R_{AC}. Only the threads that share a register can call that register's `compareAndSet()` method or read its value.

- Sketch an impossibility proof why this construction cannot be used to solve consensus between A, B, and C.
- Assume now that A, B, and C can apply a double `compareAndSet()` to both registers at once they share with the other threads. Is it now possible to solve consensus?

2 Sticky Bits

Objects of the `StickyBit` class have three possible states: $\perp$, 0, 1, initially $\perp$. A call to `write(v)`, where v is 0 or 1, has the following effects:

- If the object's state is $\perp$, then it becomes v.
- If the object's state is 0 or 1, then it is unchanged.

A call to `read()` returns the object's current state.

1. Show that such an object can solve wait-free binary consensus (that is, all inputs are 0 or 1) for any number of threads.
2. Show that an array of $\log_2 m$ StickyBit objects with atomic registers can solve wait-free consensus for any number of threads when there are m possible inputs. (Hint: You need to give each thread one single-writer, multi-reader atomic register.)