Concurrent systems

Last Updated December 2, 2022

Concepts

Correct - A process is correct if it takes infinitely many steps. **Progress** - Completes infinitely many operations. **Liveness** - The operation eventually returns something. **Safety** - The operation never returns anything incorrect (an ad-hoc rule).

Liveness Properties

Deadlock-free (DF) - If every process is correct, some process makes progress.

Starvation-free (SF) - If every process is correct, every process makes progress.

Lock-free / non-blocking (LF) - Some correct process makes progress (in a finite number of steps).

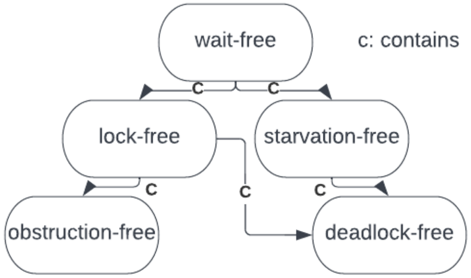
Wait-free (WF) - Every correct process makes progress (in a finite number of steps).

Obstruction-free (OF) - Every process makes progress if it executes in isolation from some point (it is the only correct process).

Periodic table of liveness properties

	Independent non-blocking (finite steps)	Dependent non-blocking	Dependent blocking (infinite steps)
every process makes progress	wait-freedom	obstruction-freedom	starvation-freedom
some process makes progress	lock-freedom	?	deadlock-freedom

Relations between liveness properties



Register

Dimensions

- Value ranges: The set of values that can be stored in the register.
- Access pattern: The number of processes that can read or write the register.
    - single reader: 1R
    - multi reader: MR
    - single writer: 1W
    - multi writer: MW
- Concurrent behavior: The correctness guarantees ensured when the register is accessed concurrently.
    - Atomicity: linearizable
    - Safety: (single writer) If write does not overlap, return last written value, otherwise return any value in its range.
    - Regularity: (single writer) If write does not overlap, return last written value, otherwise return value written or the precedent.

Frank Facundo