Concurrent systems

Last Updated December 2, 2022

Concepts

Liveness - The operation eventually returns something.

Safety - The operation never returns anything incorrect (an ad-hoc rule).

Correct - Safety and Liveness

- In out context: A process that never **fails** (stops taking steps) in the middle of middle of its operation is called **correct**. We tipically assume that a correct process invokes infinitely many operations, so a process is correct if it takes infinitely many steps.

Progress - Non blocked, process completes operations in a finite amount of time.

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

Independent: Independent of concurrency.

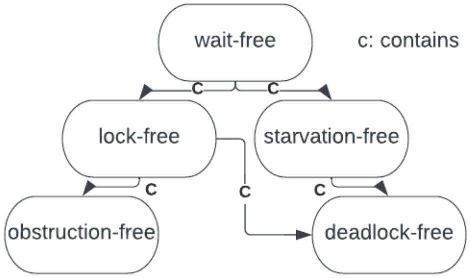
Non-blocking: At least one proccess make progress.

Independent non-blocking: Even in concurrency, at least one proccess make progress.

Dependent non-blocking: When there is no concurrency, at least one proccess make progress.

Dependent blocking: When there is no concurrency, it is not guaranteed that at least one proccess make progress.

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