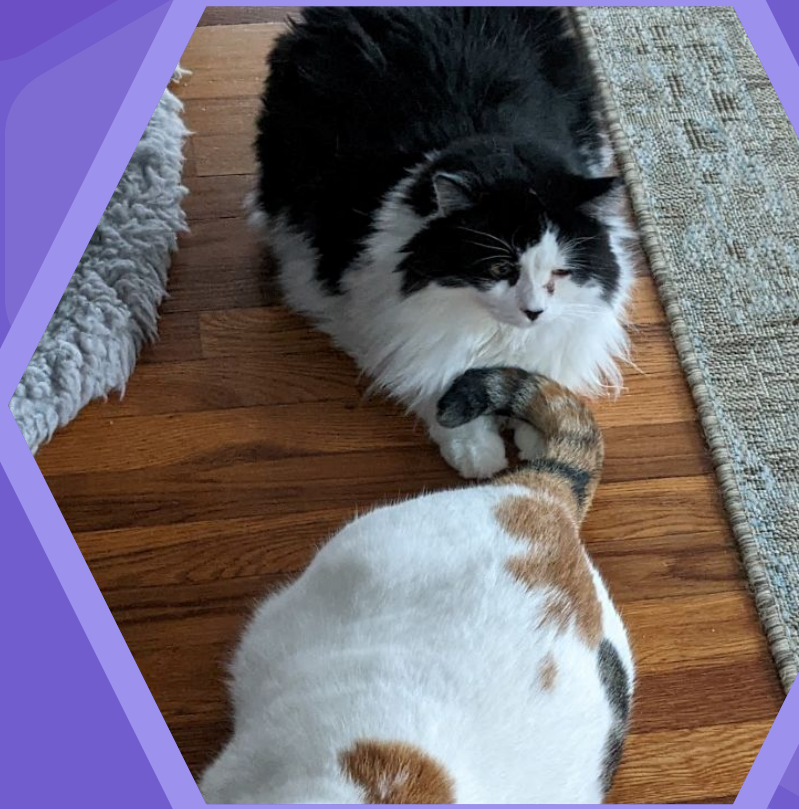
Linear Temporal Logic



AC Liveness Requirements

Safety: nothing bad ever happens

Liveness: something good eventually happens

Eventually reaching the desired temperature (as long as desired temperature hasn't changed and the system is on)

Eventually a message gets sent (or saying messages keep being sent eventually)

*How would you **monitor** that a liveness requirement is fulfilled?*

Verifying some liveness properties

Saying something *eventually* happens is the same thing as saying that it is *not* the case that it always *doesn't* happen

Can we use invariant verification to check this?

Linear Temporal Logic (LTL)

Assume you have *some* execution trace

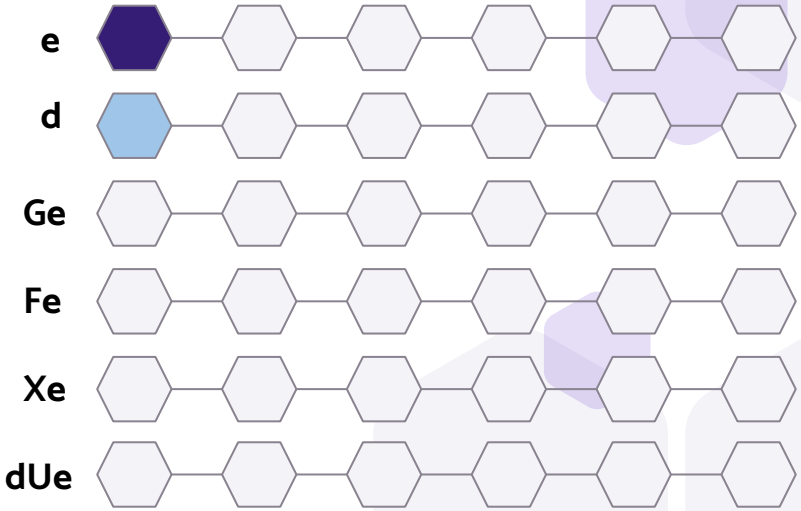
LTL operators are propositional logic operators PLUS:

G (globally/always)

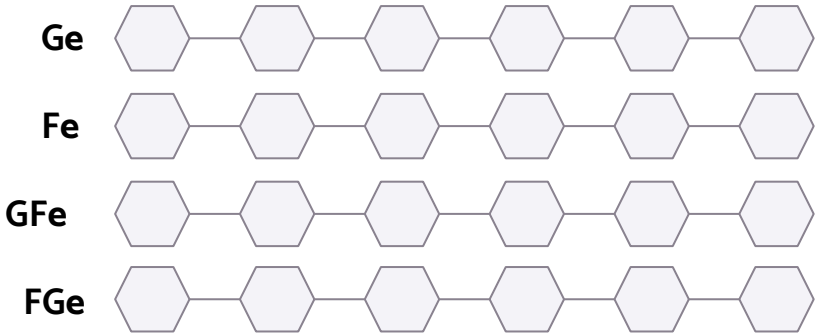
F (eventually/finally)

X (next state)

U (until)



FG vs G, GF vs F, FG vs GF



LTL examples on FSMs

Safety property is an *invariant* if property p holds for all reachable states of S

Liveness property *holds* for S if it holds for all possible traces of S

Lee/Seshia Chapter 13, exercise 2

LTL means we can specify liveness properties with F. Can we specify safety properties more easily with LTL, too?

Safety properties with LTL

Use “G” to say a property holds for every state

Can use “X” to express statefulness/history without a monitor state machine

Limits of LTL

G(even(x) -> ((X-even(x)) ∧ (XXeven(x))

But if you don't know if you started a sequence with an odd number or even number, you cannot write

(even(x) ∧ X-even(x))

Repeatability

A property *p* over the state variables of a transition system *S* is said to be *repeatable* if there exists *some* trace *q* of *S* such that *q* satisfies the recurrence LTL-formula *GFp*.

Buchi automata

Automata which “accept” a given LTL formula

(see Alur’s textbook for examples)

Automated LTL verification

Buchi automata construction can be automated (discussion in optional reading)

Compose Buchi automata of negated property with system and produce a counter-example of repeatability to prove property

- Done using a DFS + cycle detection (“nested DFS”)

More verification techniques

Automated verification

- Symbolic model checking: represents a set of states symbolically as a logic formula and does symbolic (algebraic) computation

What about timed/hybrid automata?

- Symbolic model checking for a different kind of logic (signal temporal logic)
- Assisted proof engines (differential dynamic logic)
- An active area of research!

Summary: pros/cons of verification?