EECS 219C: Formal Methods

Formal Specification / Temporal Logic: Examples Used in Lecture

Sanjit A. Seshia

EECS, UC Berkeley

Examples: Safety or Liveness?

1. “No more than one processor (in a multi-processor system) should have a cache line in write mode”

2. “The grant signal must be asserted at some time after the request signal is asserted”

3. “Every request signal must receive an acknowledge and the request should stay asserted until the acknowledge signal is received”

Examples: What do they mean?

- G F p
- F G p
- G(p → F q)
- F(p → (X X q))

Temporal Operators & Relationships

- G, F, X, U: All express properties along paths
- Can you express G p purely in terms of F, p, and Boolean operators ?
- How about G and F in terms of U and Boolean operators?
- What about X in terms of G, F, U, and Boolean operators?

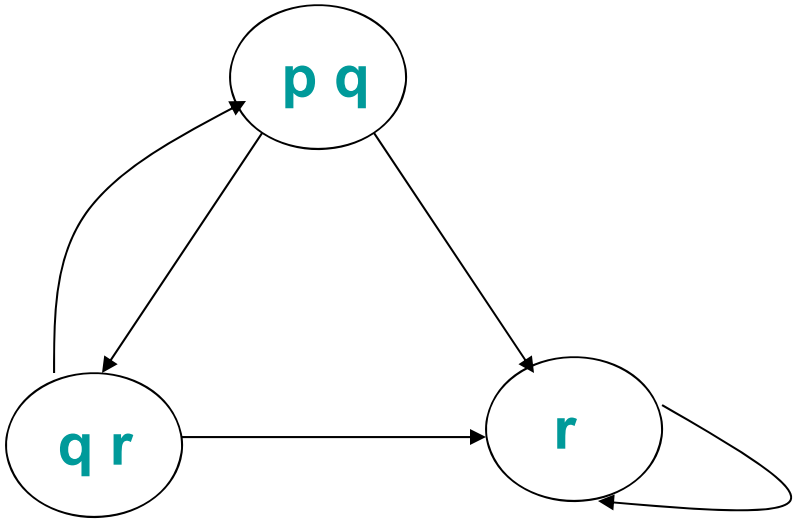
Examples in Temporal Logic

1. "No more than one processor (in a 2-processor system) should have a cache line in write mode"
   - wr_1 / wr_2 are respectively true if processor 1 / 2 has the line in write mode

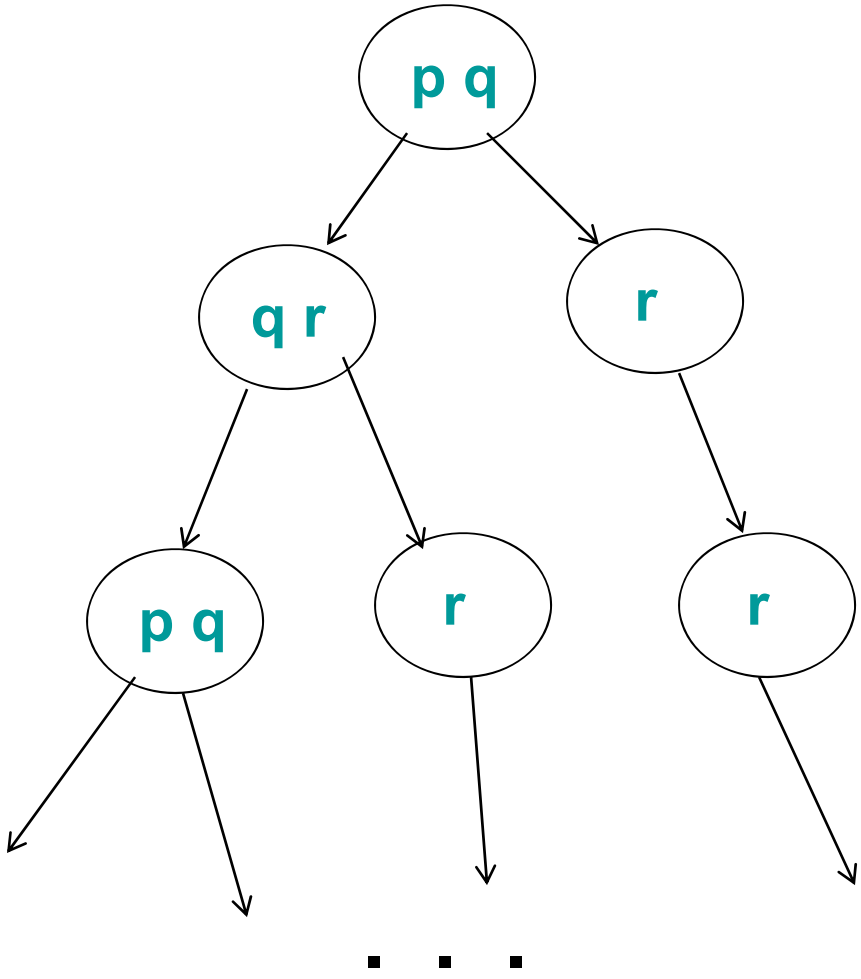
2. "The grant signal must be asserted at some time after the request signal is asserted"
   - Signals: grant, req

3. "Every request signal must receive an acknowledge and the request should stay asserted until the acknowledge signal is received"
   - Signals: req, ack

Computation Tree



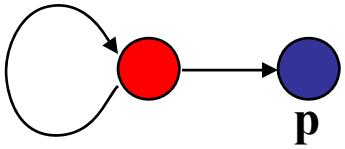
“Kripke structure”



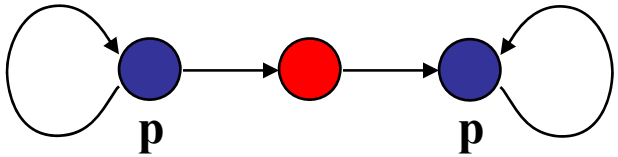
Infinite Computation Tree

CTL as a way to approximate LTL

– AG EF p is weaker than G F p **Useful for finding bugs...**



– AF AG p is stronger than F G p



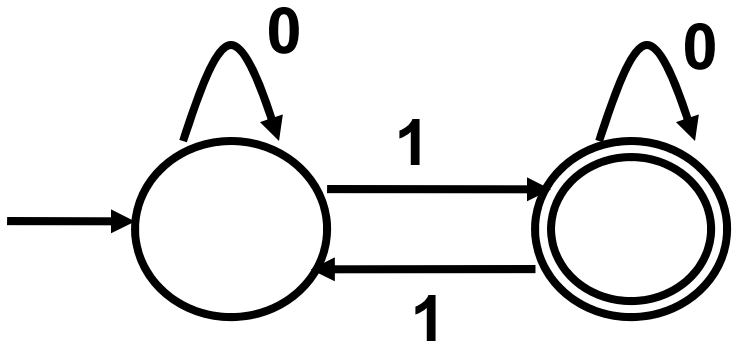
Useful for verifying correctness...

Why? And what good is this approximation?

(Absence of) Deadlock

- An oft-cited property, especially people building distributed / concurrent systems
- Can you express it in terms of
  - a property of the state graph (graph of all reachable states)?
  - a CTL property?
  - a LTL property?

Example of (Buchi)-Automaton



Language of the automaton = all finite-length binary strings with an odd number of 1s

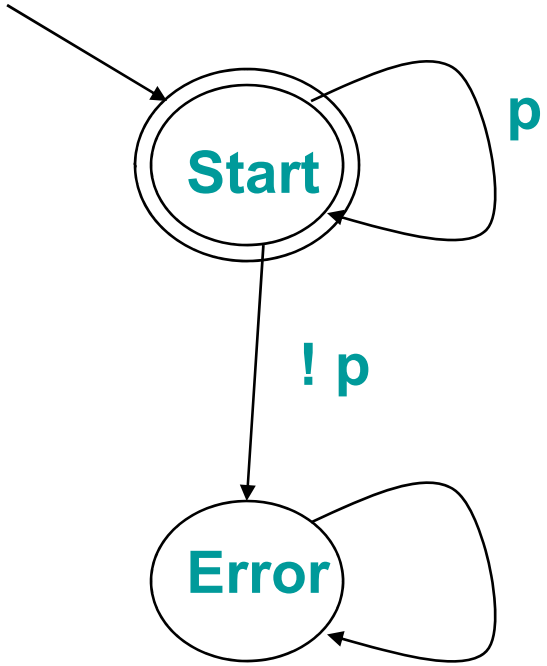
Reg. expr.: 0*1 (0 + 10*1)*

If you interpret it as a Büchi automaton over infinite words: all infinite-length binary strings with an odd parity of 1s or infinitely many 1s

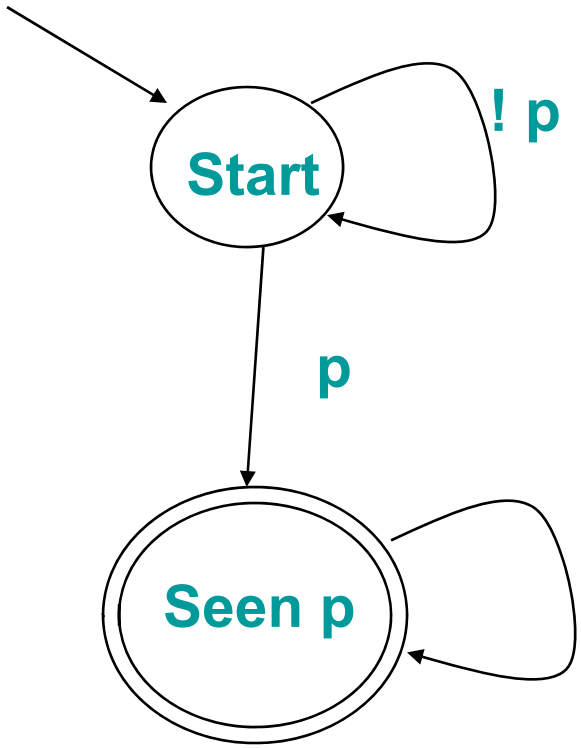
w-regular expr: $0^*1\ (0 + 10^*1)^w$

Infinitely many repetitions

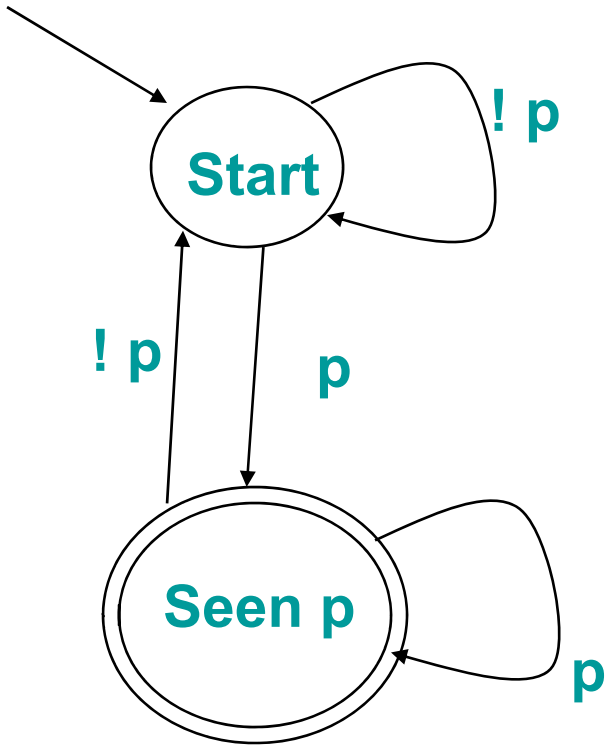
Automaton for G p, p a Boolean formula



Automaton for F p



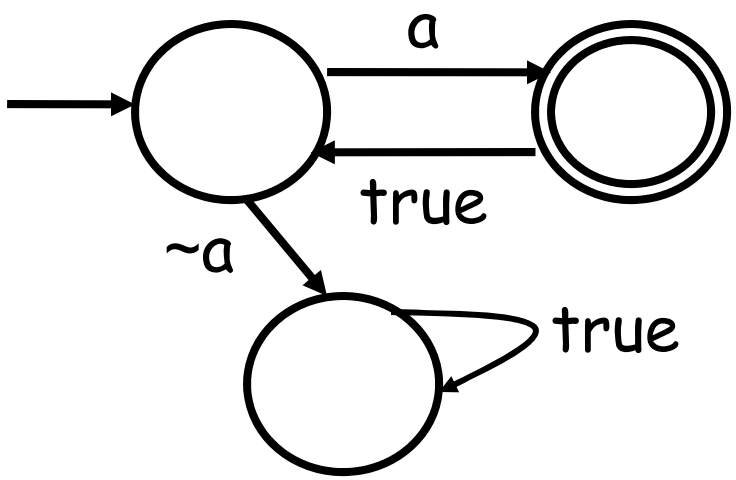
Automaton for GFp



Automaton without LTL counterpart

Automata are more expressive than LTL

What traces does the automaton below accept?



Claim: This cannot be expressed in LTL.

(How about $a \wedge G\,(a \Rightarrow X\,X\,a)$?)