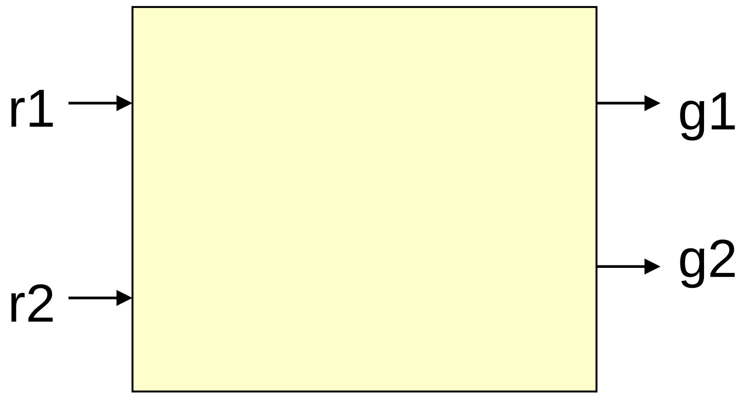
Logical Deduction: IV
Introduction to Temporal Logic

Partha P Chakrabarti

Indian Institute of Technology Kharagpur

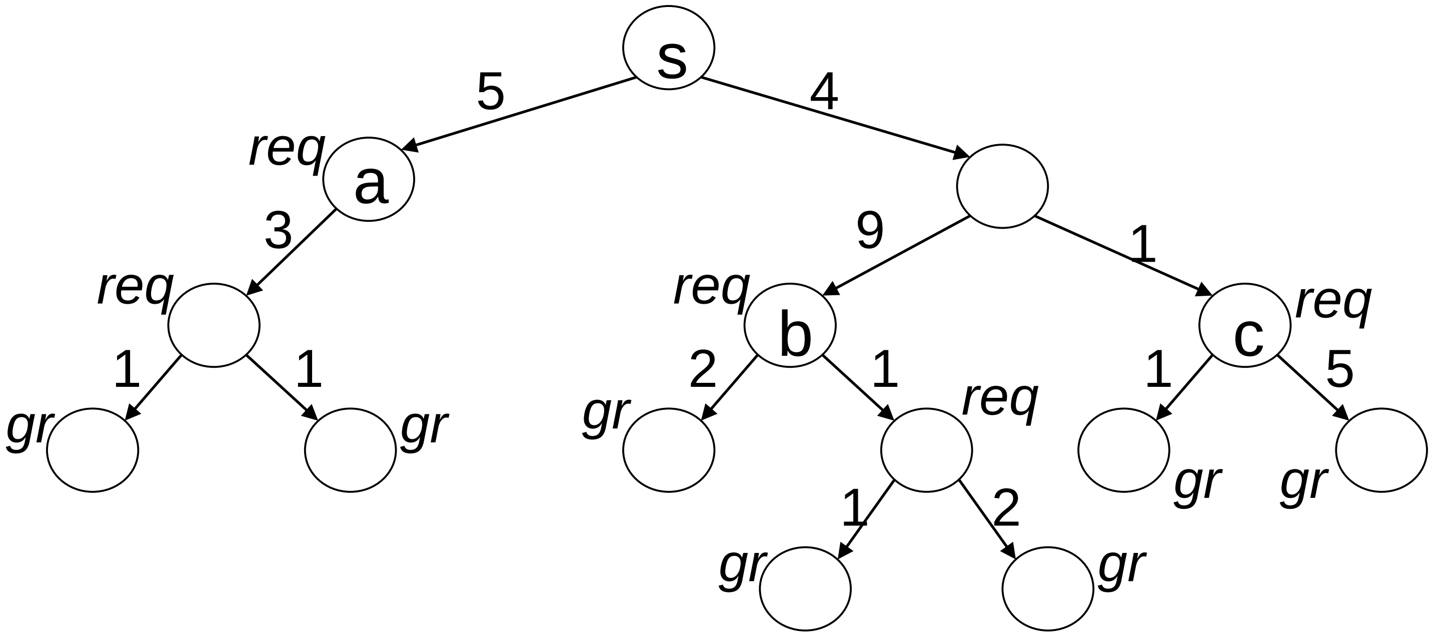
Aug-Sept 2019

Priority Arbiter: Properties



- Whenever r1 is asserted, g1 is given in the next cycle
- When r2 is the sole request, g2 comes in the next cycle
- When none of them are requesting, the arbiter parks the grant on g2
- g1 and g2 can not be true at the same time (mutual exclusion)

Analyzing Request and Grants



- From s the system always makes a request in future
- All requests are eventually granted
- Sometimes requests are immediately granted
- Requests are not always immediately granted
- Requests are held till grant is received

Timing Properties

- **Whenever a request is recorded, the grant should take place within 4 units of time.**

- **The arbiter will provide exactly 64 units of time to high-priority users in each grant.**

Car Braking

b: brakes are pressed, **a**: accelerator is pressed, **s**: car stops, **d**: car slows down

- When brakes are pressed, the car slows down in the next instant
- When no accelerator is pressed then after a while the car continuously slows down
- When brakes are constantly kept pressed and there is no accelerator pressed, the car slows down and eventually stops.

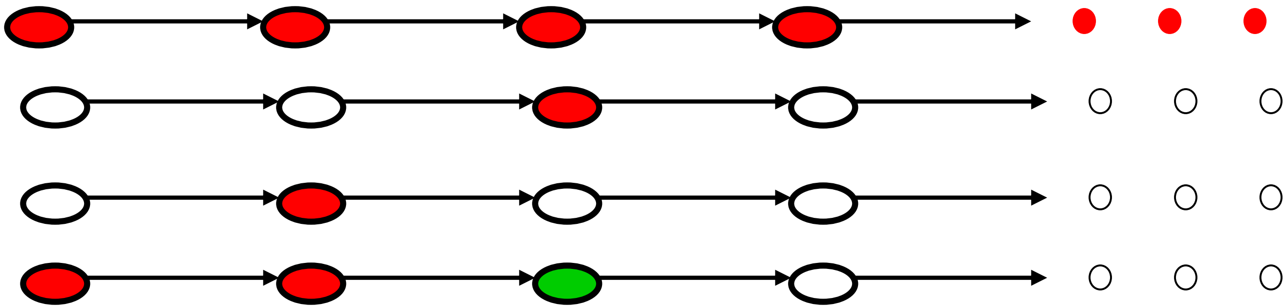
(Propositional) Temporal Logic

- A logical notation that allows to
  - specify relations in time
- Propositions are atomic
  - have definite truth values (either true or false)
- Connectives
  - Boolean operators
    - $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$
  - Temporal operators
    - G p or always p
    - F p or eventually p
    - X p or next p
    - p U q or p until q

Propositional Temporal Logic

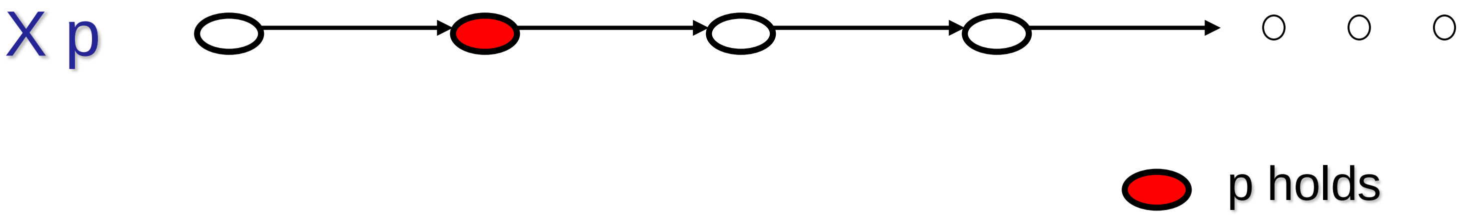
- **Temporal operators:**

  - ► **Gp**
  - ► **Fp**
  - ► **Xp**
  - ► **pUq**



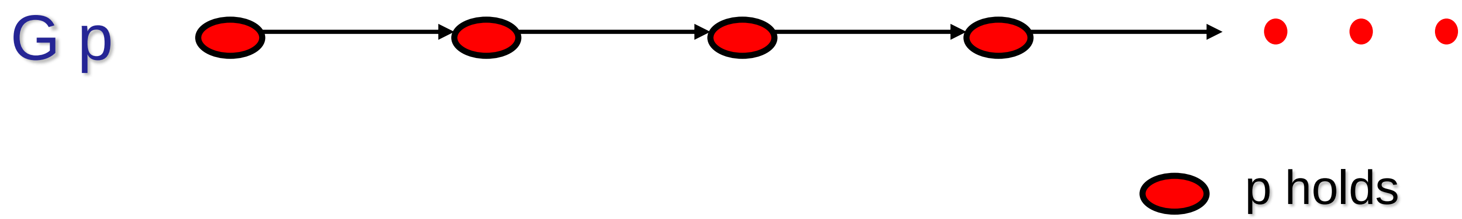
- **Path quantifiers: A** for **all** path
  E there **exists** a path

Informal Semantics



- p holds in the next state

Informal Semantics

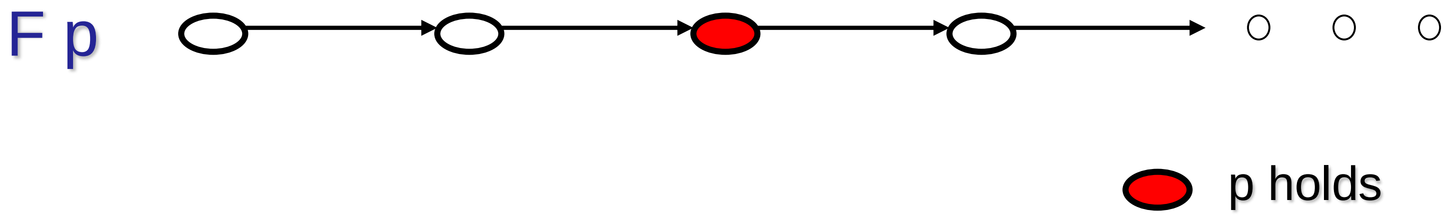


- p holds always (globally)

  alternatively

- ¬p does not hold eventually

Informal Semantics

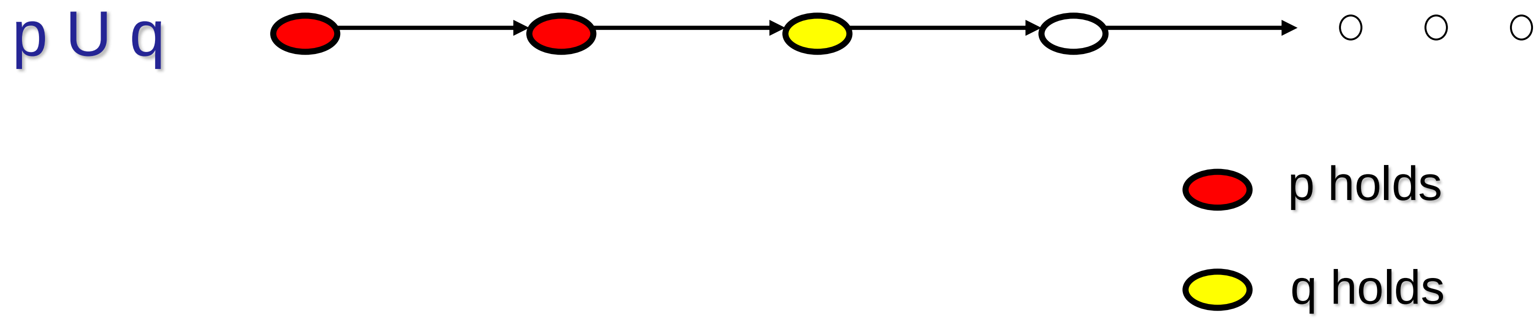


- p holds eventually (in future)

  alternatively

- ¬p does not hold always

Informal Semantics



- q holds eventually **and** p holds until q holds

Duality between Temporal Operators

G p

$\Updownarrow$

p holds always

$\Updownarrow$

¬p does not hold eventually

$\Updownarrow$

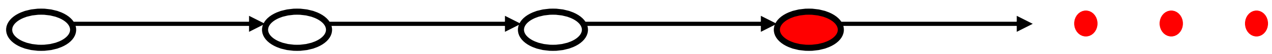
¬(¬p holds eventually)

$\Updownarrow$

¬F(¬p)

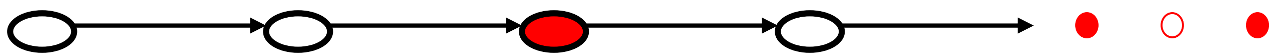
Nesting of Temporal Operators

F G p



Along the path there exists a state from which *p* will hold forever

G F p

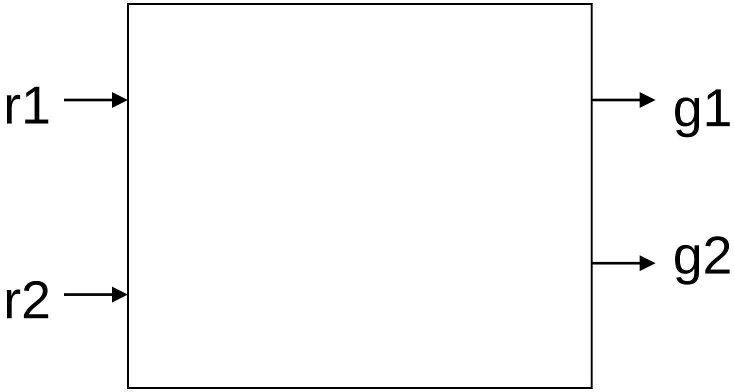


Along the path for all states there will be eventually some state where *p* holds

alternatively

Along the path p will hold *infinitely often*

Example: *Priority Arbiter*



- Either g1 or g2 is always false (mutual exclusion)

$$G[\neg g1 \vee \neg g2]$$

- Whenever r1 is asserted, g1 is given in the next cycle

$$G[\, r1 \Rightarrow Xg1 \,]$$

- When r2 is the sole request, g2 comes in the next cycle

$$G[\, (\neg r1 \wedge r2) \Rightarrow Xg2 \,]$$

- When none are requesting, the arbiter parks the grant on g2

$$G[\, (\neg r1 \wedge \neg r2) \Rightarrow Xg2 \,]$$

Temporal Logics

- **Linear Temporal Logic (LTL):**
  - LTL model checking is PSPACE complete
- **CTL:** p, q, $\neg$f, f$\wedge$g, f$\vee$g, E[fUg], A[fUg], EXf, AXf
  - All untils and next-time operators must be immediately preceded by an E or an A.
  - CTL model checking is in P.
- **CTL*:**
  - CTL without the quantifier restriction on untils and next-time operators.
  - CTL* model checking is PSPACE complete

Timing Properties

- Whenever a request is recorded, the grant should take place within 4 units of time.

$$G(\text{ request} \rightarrow F_{[0,4]} \text{ grant})$$

Automotive Properties in Temporal Logic

- When brake is applied, the car immediately decelerates

$$\mathbf{G}[\text{ brake} \Rightarrow \mathbf{X}\text{ decel }]$$

- When brake is applied, the car begins to decelerate within 200ms $\mathbf{G}[\text{ brake} \Rightarrow \mathbf{F}_{\leq 200}\text{ decel }]$
- When brake is pressed, then car decelerates within 200 milliseconds by either throttle adjustment or brake adjustment.

$$\mathbf{G}[\text{ brake} \Rightarrow \mathbf{F}_{\leq x}\text{ (throttle\_adj} \vee \text{brake\_adj) }] \wedge$$

$$\mathbf{G}[\text{ throttle\_adj} \Rightarrow \mathbf{F}_{\leq y}\text{ decel }] \wedge$$

$$\mathbf{G}[\text{ brake\_adj} \Rightarrow \mathbf{F}_{\leq z}\text{ decel }] \wedge (x+ y \leq 200) \wedge (x + z \leq 200)$$

- If brake is pressed for more than 3 seconds the car stops.

$$\mathbf{G}[\text{ brake } \mathbf{U}_{\geq 3000} \neg\text{brake} \Rightarrow \mathbf{F}_{\leq 3000}\text{ stops }]$$

Timed Temporal Logics

- Temporal logics for reasoning about timing.

RTCTL: CTL with the bounded until operator.

$$E[p\ U_{[2,6]}\ A(q\ U_{[3,7]}\ r)]$$

– RTCTL model checking is PSPACE complete

TCTL: Multiple clocks

$$z.E[p\ U\ (2 \leq Z \leq 6) \wedge w.A[q\ U\ (3 \leq w \leq 7) \wedge r]]$$

– TCTL model checking is PSPACE complete

TLTL: TCTL without E and A

– TLTL model checking is undecidable

Thank you