Linear Temporal Logic

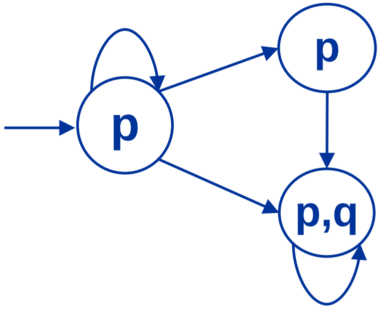
Moonzoo Kim
CS Dept. KAIST

Motivation for verification

- There is a great advantage in being able to verify the correctness of computer systems
  - This is most obvious in the case of safety-critical systems
    - ex. Cars, avionics, medical devices
  - Also applies to mass-produced embedded devices
    - ex. handphone, USB memory, MP3 players, etc
- Formal verification can be thought of as comprising three parts
  1. a system description language
  2. a requirement specification language
  3. a verification method to establish whether the description of a system satisfies the requirement specification.

Model checking

- Model checking
  - In a model-based approach, the system is represented by a model $\mathcal{M}$. The specification is again represented by a formula ϕ.
    - The verification consists of computing whether $\mathcal{M}$ satisfies ϕ $\mathcal{M} \vDash \phi$
      - Caution: $\mathcal{M} \vDash \phi$ represents satisfaction, not semantic entailment
- In model checking,
  - The model $\mathcal{M}$ is a transition systems and
  - the property ϕ is a formula in temporal logic
    - ex. $\Box$ p, $\Box$ q, $\Diamond$ q, $\Box \Diamond$ q



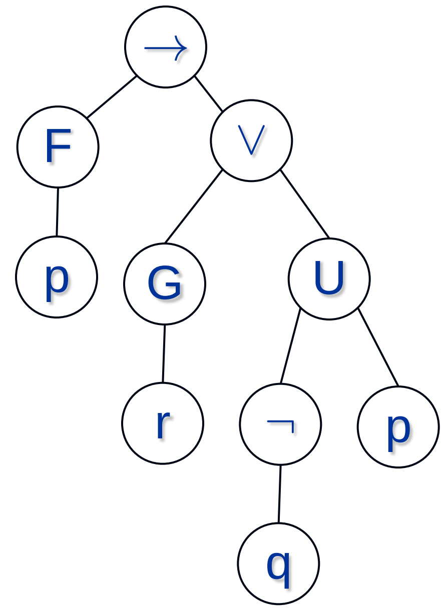
Linear time temporal logic (LTL)

- LTL models time as a sequence of states, extending infinitely into the future
  - sometimes a sequence of states is called a computation path or an execution path, or simply a path
- Def 3.1 LTL has the following syntax
  - ϕ ::= T | $\perp$ | p | $\neg\, \phi$ | $\phi \wedge \phi$ | $\phi \vee \phi$ | $\phi \rightarrow \phi$

    | X ϕ | F ϕ | G ϕ | ϕ U ϕ | ϕ W ϕ | ϕ R ϕ

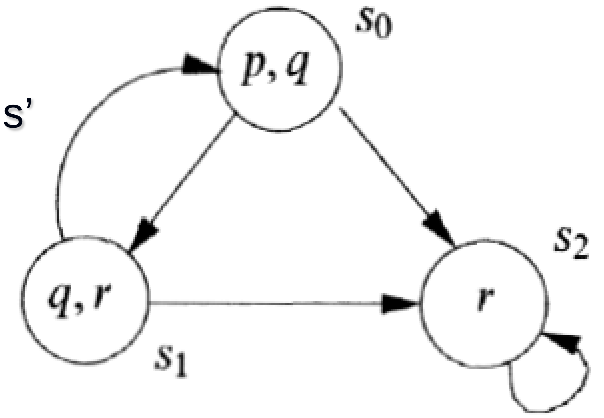
    where p is any propositional atom from some set Atoms
  - Operator precedence
    - the unary connectives bind most tightly. Next in the order come U, R, W, $\wedge$, $\vee$, and $\rightarrow$

F p $\rightarrow$ G r $\vee$ $\neg$ q U p



Semantics of LTL (1/3)

- Def 3.4 A transition system (called model) $\mathcal{M}$ = (S, $\rightarrow$, L)
  - a set of states S
  - a transition relation $\rightarrow$ (a binary relation on S)
    - such that every s $\in$ S has some s' $\in$ S with s $\rightarrow$ s'
  - a labeling function L: S $\rightarrow$ $\mathcal{P}$ (Atoms)
- Example
  - $S=\{s_0,s_1,s_2\}$
  - $\rightarrow=\{(s_0,s_1),(s_1,s_0),(s_1,s_2),(s_0,s_2),(s_2,s_2)\}$
  - $L=\{(s_0,\{p,q\}),(s_1,\{q,r\}), (s_2,\{r\})\}$

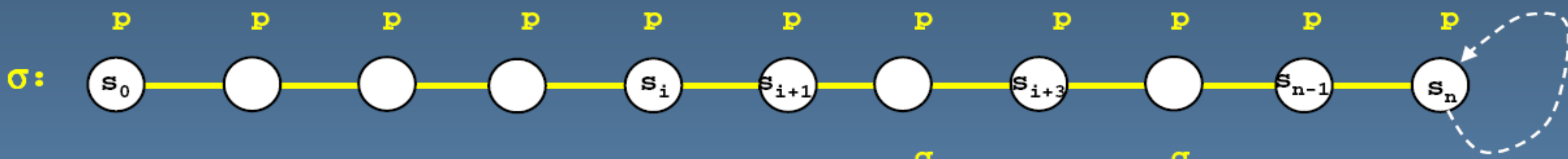


- Def. 3.5 A path in a model $\mathcal{M}$ = (S, $\rightarrow$, L) is an infinite sequence of states s_{i_1}, s_{i_2}, s_{i_3},… in S s.t. for each j$\geq$ 1, $s_{i_j} \rightarrow s_{i_{j+1}}$. We write the path as $s_{i_1} \rightarrow s_{i_2} \rightarrow$ …
  - From now on if there is no confusion, we drop the subscript index i for the sake of simple description
- We write π^i for the suffix of a path starting at s_i.
  - ex. π^3 is $s_3 \rightarrow s_4 \rightarrow$ …

Semantics of LTL (2/3)

- Def 3.6 Let $\mathcal{M}$ = (S, $\rightarrow$, L) be a model and π = $s_1 \rightarrow \ldots$ be a path in $\mathcal{M}$. Whether π satisfies an LTL formula is defined by the satisfaction relation $\vDash$ as follows:
  - Basics: $\pi \vDash \top$, $\pi \nvDash \perp$, $\pi \vDash$p iff p $\in$ L(s_1) , $\pi \vDash \neg\phi$ iff $\pi \nvDash \phi$
  - Boolean operators: $\pi \vDash$ p $\wedge$ q iff $\pi \vDash$ p and $\pi \vDash$ q
    - similar for other boolean binary operators
  - $\pi \vDash$ X ϕ iff $\pi^2 \vDash \phi$ (next $\circ$)
  - $\pi \vDash$ G ϕ iff for all i $\geq$ 1, $\pi^i \vDash \phi$ (always $\Box$)
  - $\pi \vDash$ F ϕ iff there is some i $\geq$ 1, $\pi^i \vDash \phi$ (eventually $\Diamond$)
  - $\pi \vDash \phi$ U ψ iff there is some i $\geq$ 1s.t. $\pi^i \vDash \psi$ and for all j=1,…,i-1 we have $\pi^j \vDash \phi$ (strong until)
  - $\pi \vDash \phi$ W ψ iff either (weak until)
    - either there is some i $\geq$ 1 s.t. $\pi^i \vDash \psi$ and for all j=1,…,i-1 we have $\pi^j \vDash \phi$
    - or for all k $\geq$ 1 we have $\pi^k \vDash \phi$
  - $\pi \vDash \phi$ R ψ iff either (release)
    - either there is some i $\geq$ 1 s.t. $\pi^i \vDash \phi$ and for all j=1,…,i we have $\pi^j \vDash \psi$
    - or for all k $\geq$ 1 we have $\pi^k \vDash \psi$

examples



[]p is satified at all locations in σ

<>p is satisfied at all locations in σ

[]<>p is satisfied at all locations in σ

<>q is satisfied at all locations except s_{n-1} and s_n

Xq is satisfied at s_{i+1} and at s_{i+3}

pUq (strong until) is satisfied at all locations except s_{n-1} and s_n

<>(pUq) (strong until) is satisfied at all locations except s_{n-1} and s_n

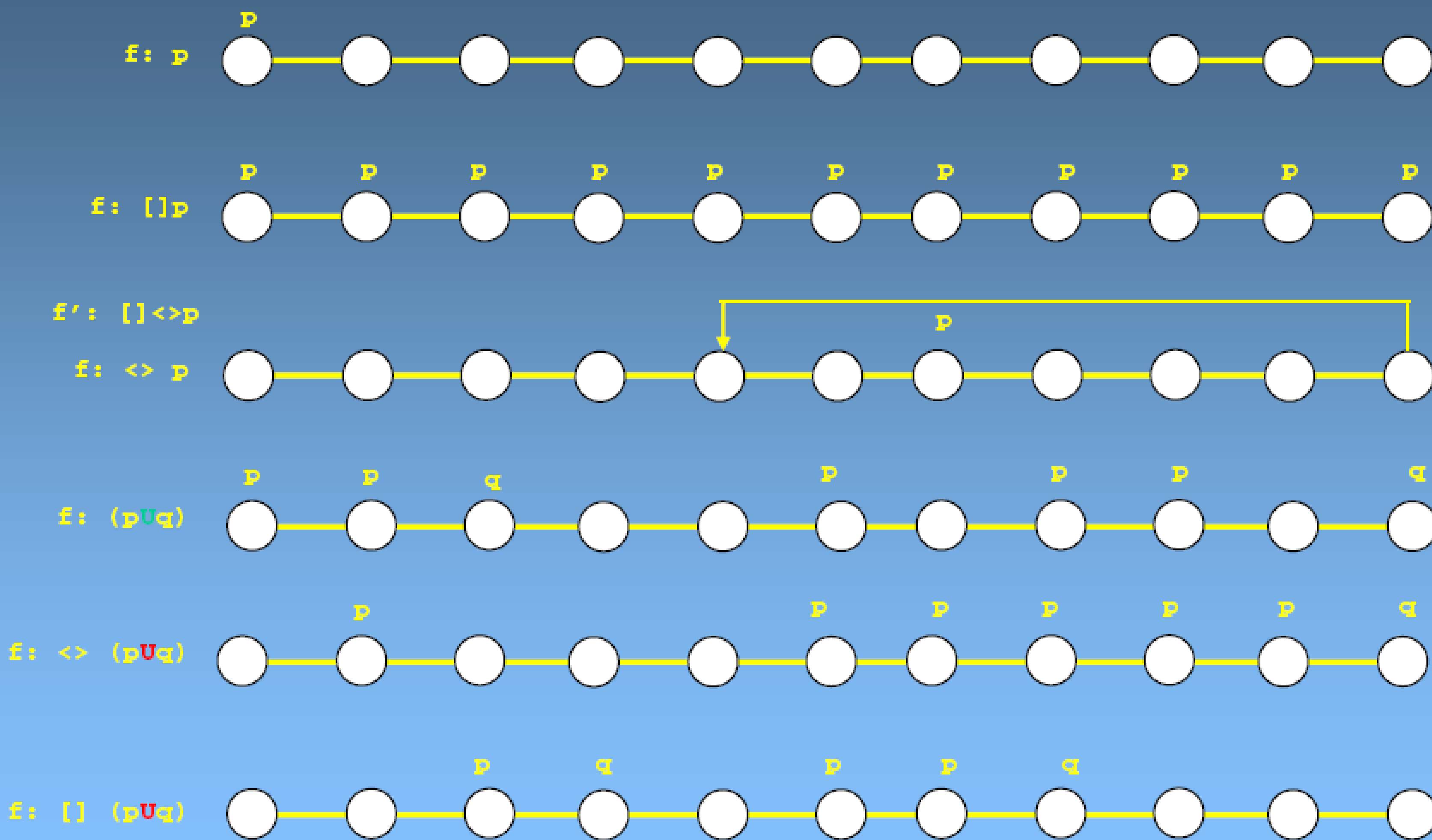
<>(pUq) (weak until) is satisfied at all locations

[]<>(pUq) (weak until) is satisfied at all locations

in model checking we are typically only
interested in whether a temporal logic formula
is satisfied for all runs of the system, starting
in the initial system state (that is: at s_0)

slide quoted from Caltech 101b.2 “Logic Model Checking” by Dr.G.Holzmann

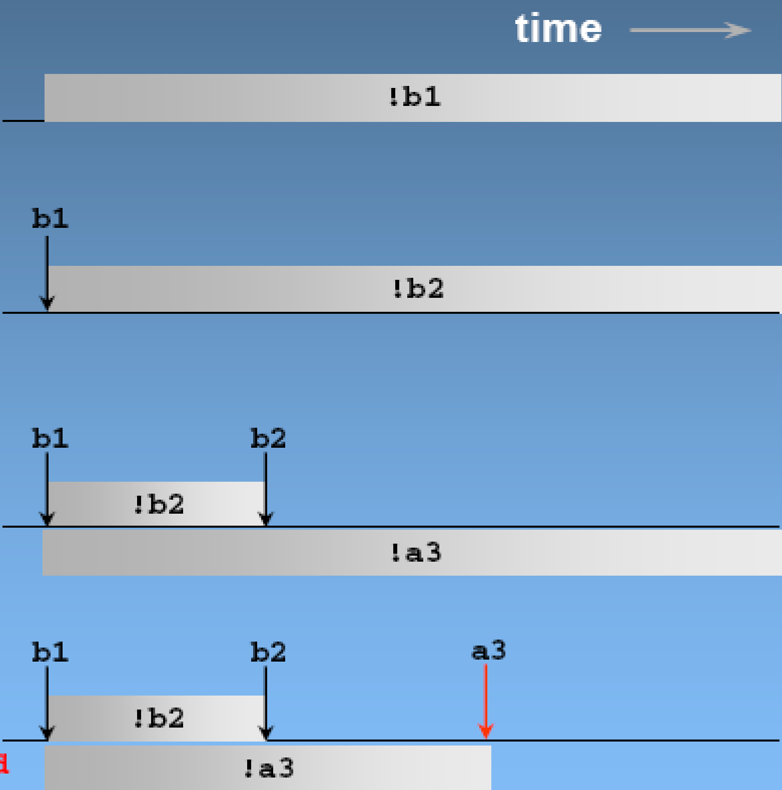
visualizing LTL formulae



slide quoted from Caltech 101b.2 “Logic Model Checking” by Dr.G.Holzmann

interpreting formulae...

LTL: (<>(b1 && (!b2 U b2))) -> []!a3



1. suppose b1 never becomes true
   (p->q) means (!p v q)
   the formula is *satisfied!*

2. b1 becomes true, but not b2
   the formula is *satisfied!*

3. b1 becomes true, then b2
   but not a3
   the formula is *satisfied*

4. b1 becomes true, then b2, then a3
   the formula is *not satisfied*
   i.e., the property is violated

slide quoted from Caltech 101b.2 "Logic Model Checking" by Dr.G.Holzmann

another example

`LTL: (<>b1) -> (<>b2)`

1. b1 never becomes true

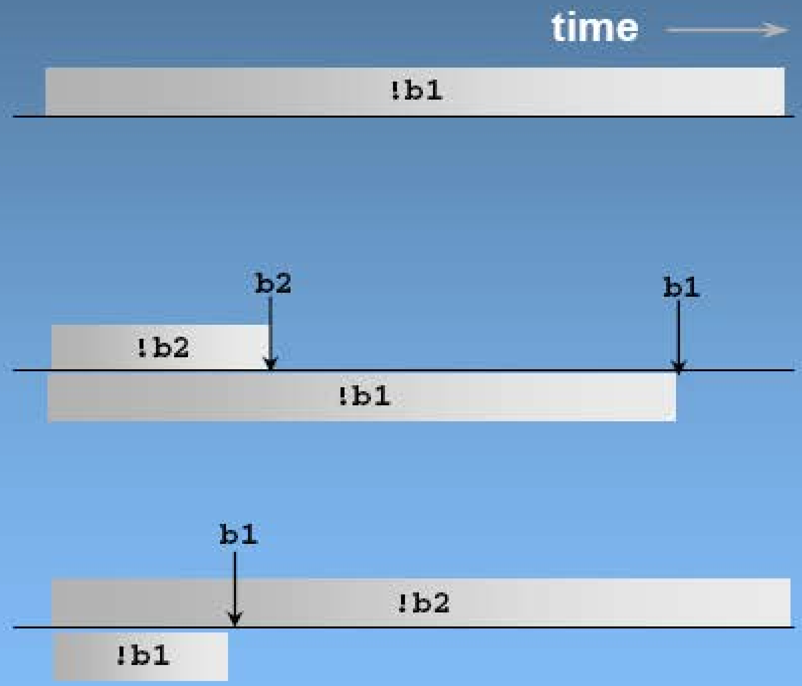
   formula satisfied

2. b1 and b2 both become true

   formula satisfied

3. b1 becomes true but not b2

   formula *not* satisfied
   the property is *violated*



slide quoted from Caltech 101b.2 “Logic Model Checking” by Dr.G.Holzmann