Temporal Logic

Moonzoo Kim
CS Division of EECS Dept.
KAIST

moonzoo@cs.kaist.ac.kr
http://pswlab.kaist.ac.kr/courses/cs402-07

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. $\square$ p, $\square$ q, $\diamondsuit$ q, $\square \diamondsuit$ q
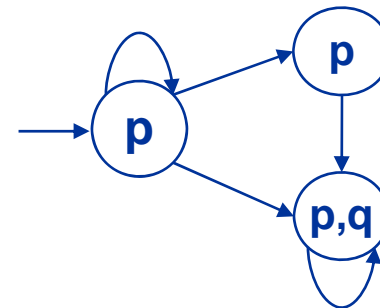
Linear time temporal logic (LTL)

- 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
    - $\phi ::= \top \mid \bot \mid p \mid \neg\,\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi$
      $\mid X\,\phi \mid F\,\phi \mid G\,\phi \mid \phi\,U\,\phi \mid \phi\,W\,\phi \mid \phi\,R\,\phi$
        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 $\rightarrow$

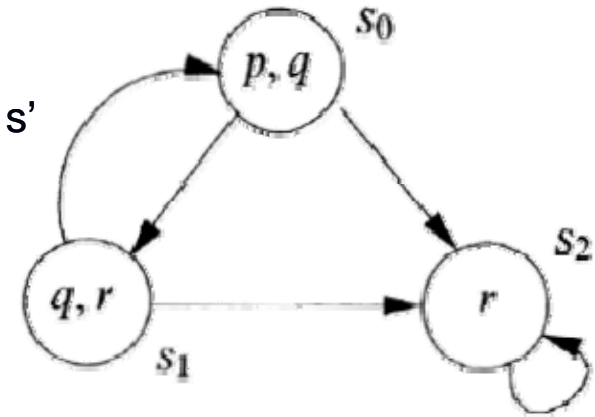$$\boxed{F\,p \rightarrow G\,r \vee \neg\,q\,U\,p}$$
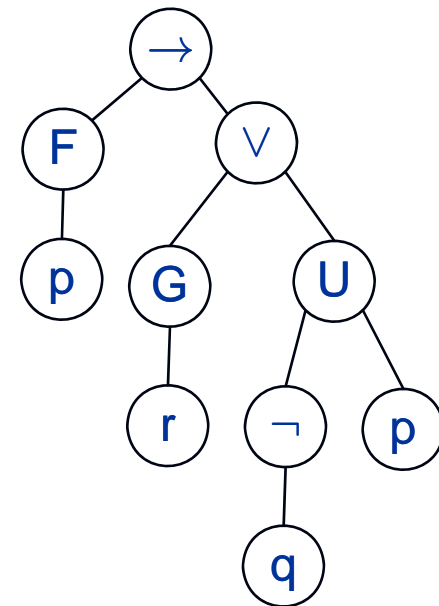
Semantics of LTL (1/3)

- 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 $s_{i_1}, s_{i_2}, s_{i_3},\ldots$ in S s.t. for each $j\geq 1$, $s_{i_j} \rightarrow s_{i_{j+1}}$. We write the path as $s_{i_1} \rightarrow s_{i_2} \rightarrow \ldots$
    - 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 $s_3 \rightarrow s_4 \rightarrow \ldots$

Semantics of LTL (2/3)

- **Def 3.6** Let $\mathcal{M} = (S, \rightarrow, L)$ be a model and $\pi = s_1 \rightarrow \ldots$ be a path in $\mathcal{M}$. Whether π satisfies an LTL formula is defined by the satisfaction relation $\models$ as follows:
  - Basics: $\pi \models \top$, $\pi \not\models \bot$, $\pi \models p$ iff $p \in L(s_1)$, $\pi \models \neg\phi$ iff $\pi \not\models \phi$
  - Boolean operators: $\pi \models p \wedge q$ iff $\pi \models p$ and $\pi \models q$
    - similar for other boolean binary operators
  - $\pi \models X \phi$ iff $\pi^2 \models \phi$ (next °)
  - $\pi \models G \phi$ iff for all $i \geq 1$, $\pi^i \models \phi$ (always □)
  - $\pi \models F \phi$ iff there is some $i \geq 1$, $\pi^i \models \phi$ (eventually ◇)
  - $\pi \models \phi \, U \, \psi$ iff there is some $i \geq 1$ s.t. $\pi^i \models \psi$ and for all $j=1,\ldots,i-1$ we have $\pi^j \models \phi$ (strong until)
  - $\pi \models \phi \, W \, \psi$ iff either (weak until)
    - either there is some $i \geq 1$ s.t. $\pi^i \models \psi$ and for all $j=1,\ldots,i-1$ we have $\pi^j \models \phi$
    - or for all $k \geq 1$ we have $\pi^k \models \phi$
  - $\pi \models \phi \, R \, \psi$ iff either (release)
    - either there is some $i \geq 1$ s.t. $\pi^i \models \phi$ and for all $j=1,\ldots,i$ we have $\pi^j \models \psi$
    - or for all $k \geq 1$ we have $\pi^k \models \psi$

examples
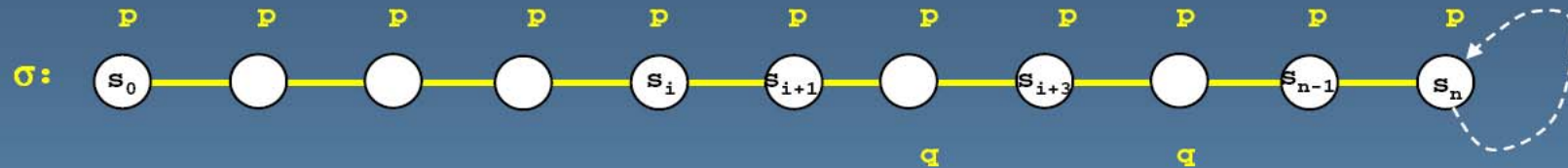

```
[]p   is satified at all locations in σ

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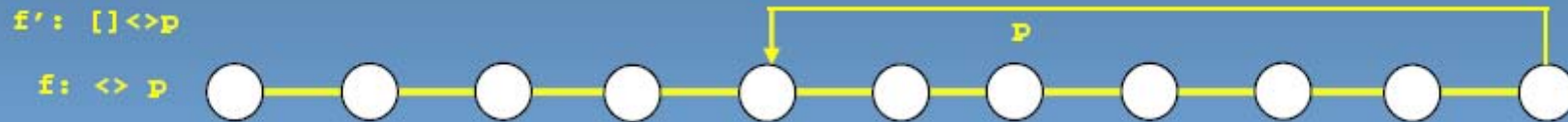
```
in model checking we are typically only
interested in whether a temporal logic formula
is satisfied for all runs of the system, starting
in the initial system state (that is: at s_0)
```

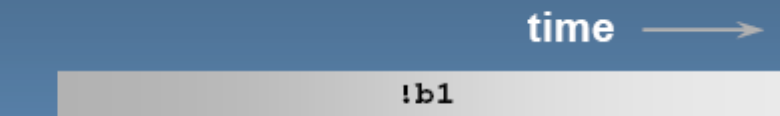*slide quoted from Caltech 101b.2 "Logic Model Checking" by Dr.G.Holzmann*

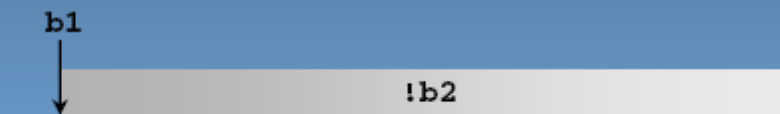slide quoted from Caltech 101b.2 "Logic Model Checking" by Dr.G.Holzmann

interpreting formulae...

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

time →

1. suppose b1 never becomes true

  `(p->q) means (!p v q)`
  `the formula is satisfied!`

  `!b1`

2. b1 becomes true, but not b2

  `the formula is satisfied!`

  b1
  `!b2`

3. b1 becomes true, then b2 but not a3

  `the formula is satisfied`

  b1   b2
  `!b2`
  `!a3`

4. b1 becomes true, then b2, then a3

  `the formula is not satisfied`
  `i.e., the property is violated`

  b1   b2   a3
  `!b2`
  `!a3`

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

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