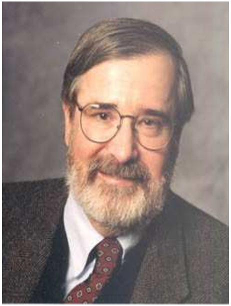
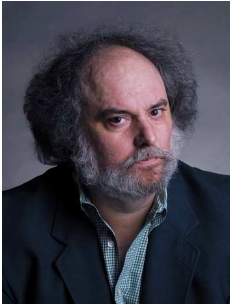
CS 228 : Logic in Computer Science

Krishna. S

Linear Temporal Logic

Model Checking

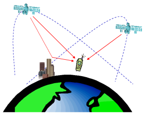



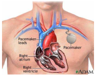

- Year 2007 : ACM confers the Turing Award to the pioneers of Model Checking: Ed Clarke, Allen Emerson, and Joseph Sifakis
- `https://amturing.acm.org/award_winners/clarke_116796`

Model checking

- Model checking has evolved in last 25 years into a widely used verification and debugging technique for software and hardware.
- Model checking used (and further developed) by companies/institutes such as IBM, Intel, NASA, Cadence, Microsoft, and Siemens, and has culminated in many freely downloadable software tools that allow automated verification.

What is Model Checking?

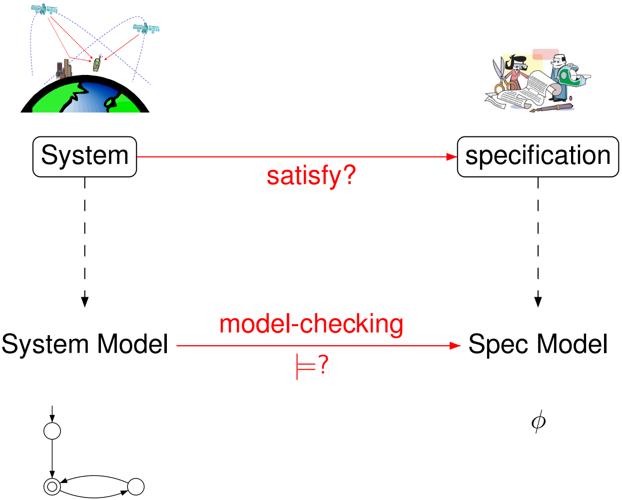




System → satisfy? → specification

good/bad properties

What is Model Checking?



Model Checker as a Black Box

- Inputs to Model checker : A finite state system M, and a property P to be checked.
- Question : Does M satisfy P?
- Possible Outputs
  - Yes, M satisfies P
  - No, here is a counter example!.

What are Models?

Transition Systems

- States labeled with propositions Model is in the state if it satisfies those propos
- Transition relation between states if certain conditions are satisfied then only transitions will occur.
- Action-labeled transitions to facilitate composition

What are Properties?

Example properties

- Can the system reach a deadlock?
- Can two processes ever be together in a critical section?
- On termination, does a program provide correct output?

Notations for Infinite Words

- Σ is a finite alphabet
- Σ^* set of finite words over Σ
- An infinite word is written as $\alpha = \alpha(0)\alpha(1)\alpha(2)\ldots$, where $\alpha(i) \in \Sigma$
- Such words are called ω-words
- a^ω, $a^7.b^\omega$ Read as w-words i.e. they are infinite words.

Transition Systems

Transition system in LTL provides a formal way to model and analyze the behavior of the system over time.

A Transition System is a tuple $(S, Act, \rightarrow, I, AP, L)$ where

- S is a set of states
- Act is a set of actions
- $s \xrightarrow{\alpha} s'$ in $S \times Act \times S$ is the transition relation What does this alpha represent
- $I \subseteq S$ is the set of initial states
- AP is the set of atomic propositions
- $L : S \rightarrow 2^{AP}$ is the labeling function
  These labels are put on the transition and they are the alphabets which are going to be true in next state.

Traces of Transition Systems

Yes, in the context of formal logic and model checking, a trace is a sequence of states or events representing the evolution of a system over time. Each state in a trace can be characterized by a set of propositions (or atomic statements) that hold true at that particular state. So, the trace essentially represents a series of snapshots of the system, where each snapshot (state) is defined by the propositions that are true at that moment.

- Labels of the locations represent values of all observable propositions $\in AP$
- Captures system state
- Focus on sequences $L(s_0)L(s_1)\ldots$ of labels of locations
- Such sequences are called traces
- Assuming transition systems have no terminal states,
  - Traces are infinite words over 2^{AP}
  - Traces $\in (2^{AP})^\omega$
  - Go to the example slide and define traces

Traces of Transition Systems

The trace of a transition state in the context of a transition system represents a specific sequence of states that the system can traverse, starting from an initial state

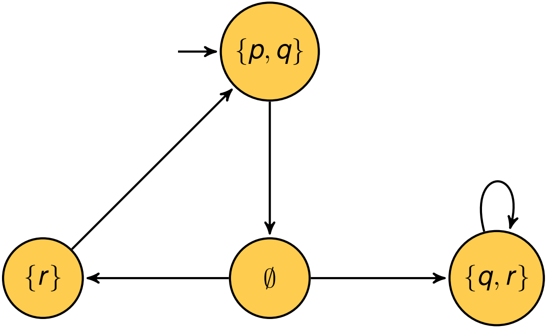
Given a transition system $TS = (S, Act, \to, I, AP, L)$ without terminal states,

- All maximal executions/paths are infinite
- Path $\pi = s_0 s_1 s_2 \ldots$, $trace(\pi) = L(s_0)L(s_1)\ldots$
- For a set Π of paths, $Trace(\Pi) = \{trace(\pi) \mid \pi \in \Pi\}$
- For a location s, $Traces(s) = Trace(Paths(s))$
- $Traces(TS) = \bigcup_{s \in I} Traces(s)$

Each state in the trace reflects the configuration of the system at that moment, along with the atomic propositions that are true in that state.

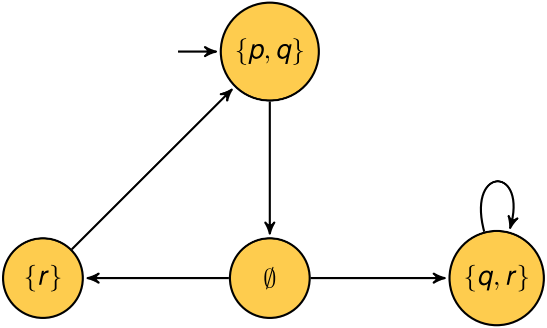
Example Traces

Doubt, how are we doing transition here.



$AP = \{p, q, r, t\}$

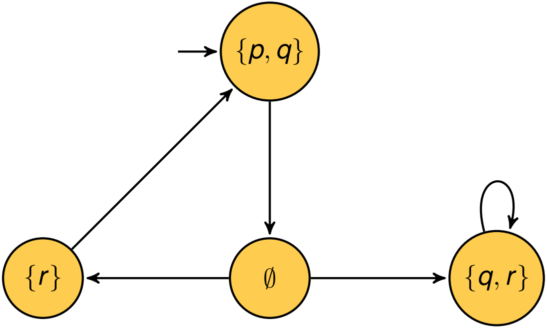
Example Traces



$AP = \{p, q, r, t\}$

- $\{p, q\}\emptyset\{q, r\}^\omega$

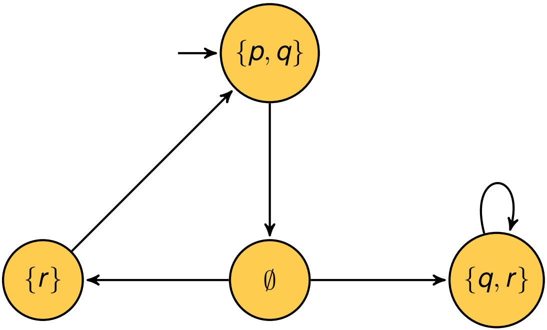
Example Traces



$AP = \{p, q, r, t\}$

- $\{p, q\}\emptyset\, \{q, r\}^\omega$
- $(\{p, q\}\emptyset\{r\})^\omega$

Example Traces



$AP = \{p, q, r, t\}$

- $\{p, q\}\emptyset\ \{q, r\}^\omega$
- $(\{p, q\}\emptyset\{r\})^\omega$
- $(\{p, q\}\emptyset\{r\})^*\ \{p, q\}\emptyset\ \{q, r\}^\omega$

Why did we put *, when we can use 'w' there.

Linear Time Properties

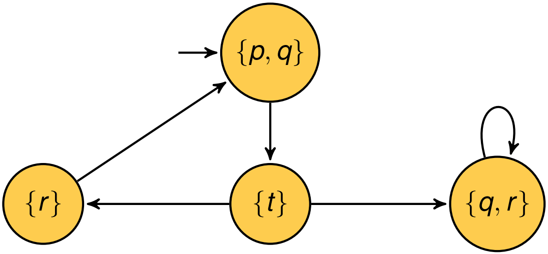
Linear time properties are specifications that describe the behavior of a system over time in a linear fashion, typically represented using temporal logic like Linear Temporal Logic (LTL).

- Linear-time properties specify traces that a TS must have
- A LT property P over AP is a subset of $(2^{AP})^\omega$ we must visit good states infinitely often.
- TS over AP satisfies a LT property P over AP

$$TS \models P \text{ iff } Traces(TS) \subseteq P$$

- $s \in S$ satisfies LT property P (denoted $s \models P$) iff $Traces(s) \subseteq P$

Specifying Traces



- Whenever p is true, r will eventually become true
  - $\{A_0 A_1 A_2 \cdots \mid \forall i \geqslant 0, p \in A_i \rightarrow \exists j \geqslant i, r \in A_j\}$
- q is true infinitely often
  - $\{A_0 A_1 A_2 \cdots \mid \forall i \geqslant 0, \exists j \geqslant i, q \in A_j\}$
- Whenever r is true, so is q
  - $\{A_0 A_1 \cdots \mid \forall i \geqslant 0, r \in A_i \rightarrow q \in A_i\}$

this implies condition puts an 'if' before the statement.

Syntax of Linear Temporal Logic

Given AP, a set of propositions,

Syntax of Linear Temporal Logic

Given *AP*, a set of propositions,

- Propositional logic formulae over *AP*
  - $a \in AP$ (atomic propositions)
  - $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$

Syntax of Linear Temporal Logic

Given *AP*, a set of propositions,

- Propositional logic formulae over *AP*
  - $a \in AP$ (atomic propositions)
  - $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$
- Temporal Operators
  - $\bigcirc\varphi$ (Next φ) We want phi to be true in the next state
  - $\varphi\,\mathsf{U}\psi$ (φ holds until a ψ-state is reached)
- LTL : Logic for describing LT properties Linear properties over time.

Semantics (On the board)

LTL formulae φ over AP interpreted over words $w \in \Sigma^\omega$, $\Sigma = 2^{AP}$,
$w \models \varphi$