Temporal Logic

19CSE205 : PROGRAM REASONING

Dr. Swaminathan J

Assistant Professor

Department of Computer Science and Engineering

AMRITA VISHWA VIDYAPEETHAM

Jul - Dec 2020

Contents

1 Temporal Property

2 Temporal Logic

3 Linear Temporal Logic

4 LTL Syntax

5 LTL Semantics

6 Combining properties

Temporal properties are expressed in temporal logic

- Just like propositional or predicate logic

Propositional Logic

- Sky is blue
- Sun rises in the north
- $\pi = 3.14$

Predicate Logic

- Some people are dishonest
- All even numbers are composite
- $a + b = 10$

Temporal Logic

- It is raining
- Stack is empty
- $x = 5$
- Philosopher 1 has one fork

Most systems are dynamic
Truth value depends on time

Wikipedia definition

- Temporal logic is any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time.

Types of temporal logic

- Linear Temporal Logic (LTL)
- Computational Tree Logic (CTL)
- Interval Temporal Logic (ITL)
- Hennesey-Milner Logic (HML)
- CTL* Generalization of CTL and LTL
- ...

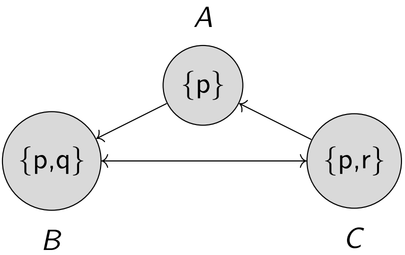
In LTL time is discretized

- Is a sequence of moments
- Has a beginning: t_0
- Runs infinitely into the future
- $t_0, t_1, t_2, ...$

A run of a finite state model

- Is a sequence of states
- Has a starting state S_0
- Does not terminate
- $S_0, S_1, S_2, ...$

Recall the finite state model



Different runs of this model

- TIME: t_0 t_1 t_2 t_3 ....
- Run 1: A B C A B C A B C ...
- Run 2: A B C B C B C B C ...
- (Infinite runs are possible)

Each run is a sequence of states

Boolean operators

- $\neg$ Negation
- $\wedge$ Conjunction
- $\vee$ Disjunction
- $\oplus$ Exclusive Or
- $\rightarrow$ Implication
- $\leftrightarrow$ Double Implication

Temporal operators

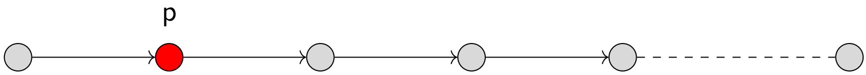
- X Next
- F Finally
- G Globally
- U Until
- W Weak Until
- R Release

An LTL formula evaluates to TRUE or FALSE.

Next we look at the semantics.

The neXt operator verifies if a property p is true in the next state.

- Unary operator
- Syntax: X p
- Alternately O p (circle) is used

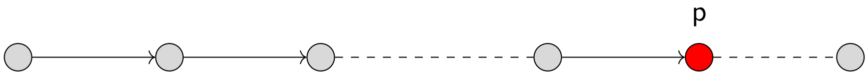


Example:

- Scenario: Dining Philosophers
- Start state: [T,T,T,T,T]
- p_1: A philosopher is hungry. p_2: A philosopher is eating.
  X p_1

The Finally operator verifies if a property p is true eventually.

- Unary operator
- Syntax: F p
- Alternately $\Diamond$ p (diamond) is used



Example:

- Scenario: Dining Philosophers
- Start state: [T,T,T,T,T]
- p_1: A philospher is eating. p_2: Phil 1 and 2 are eating.
  F p_1

The Globally operator verifies if a property p is true always.

- Unary operator
- Syntax: G p
- Alternately □ p (box) is used

Example:

- Scenario: Dining Philosophers
- Start state: [T,T,T,T,T]
- p_1: Phil 1 and 2 are not eating. p_2: A philosopher is thinking.
  G p_1

The Until operator verifies if a property p is true until property q becomes true. Property q can be true in the current or future state.

- Binary operator
- Syntax: p U q

Example:

- Scenario: Dining Philosophers
- Start state: [T,T,T,T,T]
- p: Philosopher 1 is thinking. q: Philosopher 1 is hungry.
  p U q

The Weak until operator verifies if a property p is true until property q becomes true. Property q need not become true at all in which case p remains true forever.

- Binary operator
- Syntax: p W q

Example:

- Scenario: Dining Philosophers
- Start state: [T,T,T,T,T]
- p: Philosopher 1 is thinking. q: Philosopher 1 is hungry.
  p W q Question: What if Philosopher 1 never gets hungry?

The Release operator verifies if a property q is true until p is true. At the point when p becomes true, q must also be true.

- Binary operator
- Syntax: p R q

Example:

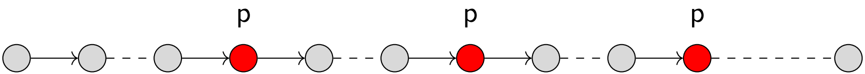
- Scenario: Dining Philosophers
- Start state: [l,T,r]
- p: Phil 1 picks up the forks. q: Phil 1 is hungry.
  p R q

1. Finally Global: F G p

- Eventually p is true forever.
- Example: Eventually the fan runs at full speed.

2. Globally Final: G F p

- Property p is true again and again
- Example: A philosopher gets to eat (implies no starvation)



3. X (p → F q)

- If p is true in the next state, q must be true eventually. If p is never true, q need not be true eventually.
- Example: If a philosopher feels hungry, he must eat eventually.

4. G (p → F q)

- Whenever p is true, q will be true eventually.
- Example: Whenever a philosopher is hungry, he will eat eventually.