

COMP4100 Lectures

Malcolm Newey

10 May, 2007

Introduction to Temporal Logic

Lecture 12 March 26, 2007

References:

Principles of Concurrent and Distributed Programming,

M. Ben-Ari, Prentice-Hall, 1990.

Temporal Logic of Programs,

Fred Kroger, Springer-Verlag, 1987.

Purpose and Scope

Why have a new logic?

- Predicate Calculus and Higher Order Logic don't deal with time.
- Z and Petri Nets can describe changes of state but one can not readily talk about what happens eventually.

Why talk about the future?

- Safety and Liveness are very important in systems
- Safety sometimes paraphrased as "Nothing bad will happen"
- Liveness sometimes paraphrased as "Something good will happen"

Sample Sentences of Linear LogicWe introduce two *modalities* written \square (always) and \diamond (eventually).

- $\square \textit{profit} > 0$ means that *profit* will be positive in all future states.
- $\diamond \textit{savings} > 10000$ means that *eventually* the variable *savings* will exceed 10,000.

Such *modal* logics were pioneered in philosophy with the operators *possibly* and *necessarily*.**What are Systems in this Context**

A system is said to follow a infinite sequence of states that are numbered from 1.

- State 1 is the initial state (and is called *init*).
- The greek letter σ is used to name the infinite sequence of states (the system's *trace*).
- The term σ_j is used to refer to the *j*th state in the sequence of states called σ .

The formula $\square P$ is a claim that *P* is true in *init* and every subsequent state.The formula $\diamond P$ is a claim that *P* will be true in *init* or some future state.

Safety and Liveness

Safety (Nothing bad will happen) is exemplified by the conditions:

- The account balance will never be negative.
- The car will not crash

An expression of safety normally has the form $\Box \neg P$

Liveness (Something good will happen) is exemplified by the conditions:

- Every dispatched message will be received.
- The system will eventually halt.

An expression of liveness normally has the form $\Diamond P$

Flavours of Temporal Logic

- This lecture will introduce *Linear Temporal Logic*.
- There are also extensions to the basic LTL.
- There is also *Branching Time* Temporal Logic.
- Fairness is an important aspect of BTTL.
- CTL etc. etc.

In all cases system behaviour is specified in terms of the properties of execution traces.

Why are we Doing This?

Model checking is of great practical significance.

- TLA (Temporal Logic of Actions) by Lamport
- TLA+ is a complete specification language
- SMV model checker(from CMU)
- Promela (PROcess MEta LAnguage) and SPIN (from Bell Labs)
- Model checking is widely used in hardware design

Primitive Ideas

- A *trace* is a sequence of states that system passes through
- We write $\sigma = \langle s_1, s_2, s_3, \dots \rangle$
- The traces are deemed infinite.
For a computation that halts in state s_j we define subsequent states to be the same as s_j (which we call *halt*).
- To indicate that some property P holds in state s_i we write $(\sigma, i) \models P$
(To start with, P can be any PC or HOL formula.)

You must remember this notation: $(\sigma, i) \models P$

Always and Ever

- We define \Box as follows:

$$(\sigma, j) \models \Box P$$

means

$$\forall k : j.. \bullet (\sigma, k) \models P$$

which, in turn, means

$$\forall k \bullet k \geq j \Rightarrow (\sigma, k) \models P$$

- We read \Box as “Forever,” “From now on,” “Always” or “Henceforth”.

Initial State is Special

- When we write a predicate without reference to a particular state, we mean it to apply to the initial state.
- That is, $(\sigma, 1) \models P$ is written simply P .
- Consequently, $\Box P$ means P is true in all states including the first (as we foreshadowed).

Concrete Syntax, an aside

- \Box and \Diamond bind tightly (have high precedence)
- $\Box P \rightarrow Q$ means $(\Box P) \rightarrow Q$
- $\Diamond P \vee Q$ means $(\Diamond P) \vee Q$

Eventuality

- We define \Diamond as follows:

$$(\sigma, j) \models \Diamond P$$

means

$$\exists k : j.. \bullet (\sigma, k) \models P$$

- We read \Diamond as “Eventually,” or “Sometime”.
- Of course, previous convention tells us what $\Diamond P$ means.

Multiple Modalities

\square and \diamond apply to temporal logic formulae, not just PC formulae.
Of course the definition of \models becomes more technical.

Examples:

- $\diamond\square P$ means *P will eventually be true forever.*
- $\square\diamond P$ reads *It will always be the case that P will eventually be true.*
- A simpler English reading of $\square\diamond P$ is *P will be true infinitely often.*

Some Things should Always Hold

- $\square(1 > 0)$
- *isInitial*
- $(\sigma, j) \models (j > 1 \rightarrow \neg \text{isInitial})$
- $(\sigma, 2) \models \square \neg \text{isInitial}$
- $((\sigma, j) \models \text{isFinal}) \rightarrow ((\sigma, j) \models \square \text{isFinal})$
- $\square(\neg (\text{inCritSec}(\text{sec}_1) \wedge \text{inCritSec}(\text{sec}_2)))$

Correctness

Suppose P and Q are the pre- and post-conditions for a computation with trace σ .

- Partial correctness (with respect to these predicates) is:
 $P \rightarrow \square(\text{isFinal} \rightarrow Q)$
- Total correctness is given by:
 $P \rightarrow \diamond(\text{isFinal} \wedge Q)$

The Leads-To Relation

- The formula $\square(P \rightarrow \diamond Q)$ says that whenever P becomes true then Q also becomes true, either at the same time or subsequently.
- This is read as P leads to Q .
- Causality is such a common phenomenon in systems that we have a special notation.
- $P \rightsquigarrow Q$ is defined to be $\square(P \rightarrow \diamond Q)$.

Negation in Temporal Logic

- Recall that in PC, $(\neg \forall x. P) \equiv (\exists x. \neg P)$
and $(\neg \exists x. P) \equiv (\forall x. \neg P)$
- Similar dualities apply with box and diamond
- $\Box P \equiv \neg \Diamond \neg P$
- $\Diamond P \equiv \neg \Box \neg P$

Exercises in Temporal Logic

Which of these are true? Why?

1. $\Box(p \wedge q) \rightarrow (\Box p \wedge \Box q)$
2. $(\Box p \wedge \Box q) \rightarrow \Box(p \wedge q)$
3. $\Diamond(p \wedge q) \rightarrow (\Diamond p \wedge \Diamond q)$
4. $(\Diamond p \wedge \Diamond q) \rightarrow \Diamond(p \wedge q)$
5. $\Box(p \vee q) \rightarrow (\Box p \vee \Box q)$
6. $(\Box p \vee \Box q) \rightarrow \Box(p \vee q)$
7. $\Diamond(p \vee q) \rightarrow (\Diamond p \vee \Diamond q)$
8. $(\Diamond p \vee \Diamond q) \rightarrow \Diamond(p \vee q)$

These are all the questions to ask to determine if \Box and \Diamond distribute over \wedge and \vee .