

Hoare logic and Model checking

Part II: Model checking

Lecture 8: Temporal logic

Christopher Pulte cp526

University of Cambridge

CST Part II – 2023/24

Recap

In the previous lecture, we saw how temporal models can be used to model various kinds of systems.

In this lecture, we will look at how temporal logic can be used to specify the behaviour of temporal models.

Why not use first-order logic?

Why not model time explicitly in first-order logic with equality and $<$, and have variables represent time points?

For example:

$$\forall t_1. p(t_1) \Rightarrow (\exists t_2. t_1 < t_2 \wedge q(t_2))$$

- ✓ It works.
- ✓ It has a well-understood theory.
- ✗ It is very error-prone.
- ✗ Is is very expensive to check.

Temporal logics

- LTL (linear temporal logic): this lecture
- CTL (computation tree logic): next lecture
- CTL* combining their expressivity: next lecture
- ...

When using model checking, one generally picks (a tool or language based on) either LTL or CTL.

LTL: linear temporal logic

LTL formulas describe temporal models by describing properties of the paths in the model.

LTL has a linear conception of time. It considers infinite (linear) paths through the temporal model in which each state has exactly one successor state. LTL does not “know about” the branching structure induced by the temporal model (corresponding to the possible alternative transitions out of some state).

Syntax of LTL.

Given a fixed set of atomic propositions AP ,

$\phi, \dots \in \text{PathProp} ::=$

\perp		false
\top		true
$\text{injp } p$		atomic proposition
$\neg \phi$		negation
$\phi_1 \wedge \phi_2$		conjunction
$\phi_1 \vee \phi_2$		disjunction
$\phi_1 \rightarrow \phi_2$		implication
$X \phi$		neXt
$G \phi$		Generally
$F \phi$		Future
$\phi_1 U \phi_2$		Until

We almost always omit injp .

Informal semantics of LTL

- An LTL formula is a path property.
- A temporal model satisfies an LTL formula, if all paths from the initial states satisfy the formula.

Informal semantics of LTL

- \perp : no path satisfies this property
- \top : every path satisfies this property
- $\text{injp } p$: the current state satisfies atomic proposition p
- $\neg\phi$: the path does not satisfy ϕ
- $\phi_1 \wedge \phi_2$: the path satisfies ϕ_1 and satisfies ϕ_2
- $\phi_1 \vee \phi_2$: the path satisfies ϕ_1 or satisfies ϕ_2
- $\phi_1 \rightarrow \phi_2$: if the path satisfies ϕ_1 then it satisfies ϕ_2

Informal semantics of LTL

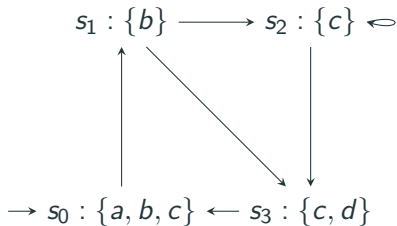
- $X \phi$: the tail of the current path satisfies ϕ
- $G \phi$: every suffix of the current path satisfies ϕ
- $F \phi$: some suffix of the current path satisfies ϕ
- $\phi_1 U \phi_2$: some suffix of the current path satisfies ϕ_2 , and all the suffixes of the current path of which that path is a suffix satisfy ϕ_1

Notation

Note: the literature sometimes uses alternative notation for the temporal operators:

- $\bigcirc\phi$ instead of $X\phi$
- $\diamond\phi$ instead of $F\phi$
- $\square\phi$ instead of $G\phi$

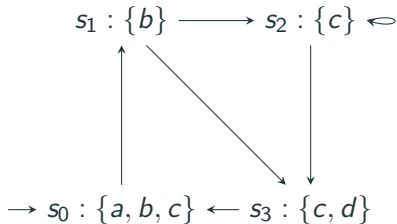
Semantics of LTL – Examples



Consider the path $\pi = s_0, s_1, s_2, s_2, s_2, \dots$. Then:

- π satisfies a
- π does not satisfy $b \wedge \text{X } c$
- π satisfies $a \rightarrow \text{F } c$
- π satisfies $\text{F } (\text{G } c)$
- π does not satisfy $\text{G } (\text{F } a)$

Semantics of LTL – Examples (2)



Consider the path $\pi = s_0, s_1, s_2, s_3, s_0, s_1, s_2, s_3, \dots$. Then:

- π does not satisfy $G (F (a \wedge b \wedge d))$
- π satisfies $(G (F (a)) \wedge G (F (b)) \wedge G (F (d)))$
- π does not satisfy $G ((c \wedge \neg a) \rightarrow X d)$
- π satisfies $G (F (c \wedge X c))$

Temporal models

A **temporal model** over **atomic propositions** AP is a left-total transition system where states are labelled with some of AP , and where some states are distinguished as initial:

$$\begin{aligned} M, \dots \in \text{TModel} &\stackrel{\text{def}}{=} \\ (S \in \text{Set}) \times &\text{states} \\ (S_0 \in \text{sub } S) \times &\text{initial states} \\ (\textcircled{1} T \textcircled{2} \in \text{relation } S S) \times &\text{transition} \\ (\ell \in (S \rightarrow \text{sub } AP)) &\text{state labelling} \end{aligned}$$

such that T is left-total:

$$\forall s \in S. \exists s' \in S. s T s'$$

Semantics of LTL

Now we make the intuition for the meaning of LTL formulas precise.

We define whether M satisfies ϕ :

$$\begin{aligned} M \models \phi &\stackrel{\text{def}}{=} \forall s \in M.S. (M.S_0 s) \Rightarrow (s \models_M^s \phi) \\ s \models_M^s \phi &\stackrel{\text{def}}{=} \left(\forall \pi \in \text{stream } M.S. \right. \\ &\quad \left. \text{IsPath } M \pi \wedge \pi 0 = s \Rightarrow \pi \models_M^p \phi \right) \end{aligned}$$

Semantics of LTL

We define whether a path π of a model M satisfies ϕ recursively.

$$\textcircled{2} \models_{\textcircled{1}}^{\mathbf{p}} \textcircled{3} \in (M \in \text{TModel}) \rightarrow \text{stream } M.S \rightarrow \text{PathProp} \rightarrow \mathbb{B}$$

We write the arguments that remain constant in the recursion in this shade of grey blue.

Semantics of LTL

$$\begin{aligned}\pi \models_M^p \top &\stackrel{def}{=} \top \\ \pi \models_M^p \perp &\stackrel{def}{=} \perp \\ \pi \models_M^p \text{injp } p &\stackrel{def}{=} M.\ell (\pi \ 0) \ p \\ \pi \models_M^p \neg \phi &\stackrel{def}{=} \neg (\pi \models_M^p \phi) \\ \pi \models_M^p \phi_1 \wedge \phi_2 &\stackrel{def}{=} (\pi \models_M^p \phi_1) \wedge (\pi \models_M^p \phi_2) \\ \pi \models_M^p \phi_1 \vee \phi_2 &\stackrel{def}{=} (\pi \models_M^p \phi_1) \vee (\pi \models_M^p \phi_2) \\ \pi \models_M^p \phi_1 \rightarrow \phi_2 &\stackrel{def}{=} (\neg (\pi \models_M^p \phi_1)) \vee (\pi \models_M^p \phi_2)\end{aligned}$$

Semantics of LTL

$$\pi \models_M^p \mathbf{X} \phi \stackrel{\text{def}}{=} (\text{tailn } M.S \ 1 \ \pi) \models_M^p \phi$$

$$\pi \models_M^p \mathbf{F} \phi \stackrel{\text{def}}{=} \exists n \in \mathbb{N}. (\text{tailn } M.S \ n \ \pi) \models_M^p \phi$$



$$\pi \models_M^p \mathbf{G} \phi \stackrel{\text{def}}{=} \forall n \in \mathbb{N}. (\text{tailn } M.S \ n \ \pi) \models_M^p \phi$$

$$\pi \models_M^p \phi_1 \mathbf{U} \phi_2 \stackrel{\text{def}}{=}$$

$$\exists n \in \mathbb{N}. \left(\left(\forall k \in \mathbb{N}. 0 \leq k < n \Rightarrow (\text{tailn } M.S \ k \ \pi) \models_M^p \phi_1 \right) \wedge \right. \\ \left. (\text{tailn } M.S \ n \ \pi) \models_M^p \phi_2 \right)$$

LTL examples: Goat puzzle

Assume AP includes such atomic propositions as

-  L : “the cabbage is on the left side of the river”, and
-  R : “the boat is on the right side of the river”.

Intended properties:

- The cabbage is never left alone with the goat, the goat never left alone with the wolf.
- Eventually all items are on the right side of the river.

LTL examples: Goat puzzle

- “The cabbage is never left alone with the goat, the goat never left alone with the wolf.”

$$\text{CabbageSafe} \stackrel{\text{def}}{=} \left(\text{flower}_L \wedge \text{goat}_L \right) \vee \left(\text{flower}_R \wedge \text{goat}_R \right) \vee \left(\text{flower}_L \wedge \text{cabbage}_R \right) \vee \left(\text{flower}_R \wedge \text{cabbage}_L \right)$$

$$\text{GoatSafe} \stackrel{\text{def}}{=} \dots$$

$$\text{Safe} \stackrel{\text{def}}{=} G (\text{CabbageSafe} \wedge \text{GoatSafe})$$

- “Eventually all items are on the right side of the river.”

$$\text{Live} \stackrel{\text{def}}{=} F \left(\text{flower}_R \wedge \text{cabbage}_R \wedge \text{goat}_R \wedge \text{wolf}_R \right)$$

A model satisfying $\text{Safe} \wedge \text{Live}$ is a solution to the puzzle.

LTL examples: elevator

Returning to the elevator example, assume AP includes such atomic propositions as:

- Open, Closed: the door is open/closed
- Up, Stay, Down: the elevator is moving upwards/staying/moving downwards
- $Call_n$: the elevator is called to level n
- Loc_n : the elevator is currently at level n

Intended properties:

- The door is not open at half-levels.
- The elevator does not lock people in.
- If the elevator is called to a level, then it eventually gets there.
- The path of the elevator is not entirely idiotic.

LTL examples: elevator

- “The door is not open at half-levels.”

$G ((Loc_{0.5} \vee Loc_{1.5}) \rightarrow Closed)$

(“The door is closed when the elevator is between levels.”)

- “The elevator does not lock people in.”

$G (F (Open))$

(“For every state along the path there is a subsequent state in which the door is open.” or “The elevator door is open infinitely often.”)

Or: $\neg (F (G (Closed)))$

LTL examples.

- “If the elevator is called to a level, then it eventually gets there.”

$G (\text{Call}_2 \rightarrow F (\text{Loc}_2))$

- Maybe: $G (\text{Call}_2 \rightarrow (\text{Call}_2 \text{ U } (\text{Loc}_2 \wedge \text{Open})))$

“If there is a call to level 2, this call is not lost until the elevator is at level 2 and has opened the door.”

- “The path of the elevator is not entirely idiotic.”
?

Caution: implication and negation.

What is the meaning of $\neg \text{ap}$ (for some atomic proposition ap); does $\neg \text{ap}$ hold for a path π where $s = \pi 0$ is not labelled with ap ?

$$\pi \models_M^p \neg \text{ap} = \neg(\pi \models_M^p \text{ap}) = \neg(M.\ell(\pi 0) \text{ ap}) = \top$$

Use of negation and implication are potentially brittle, because they can conflate two situations:

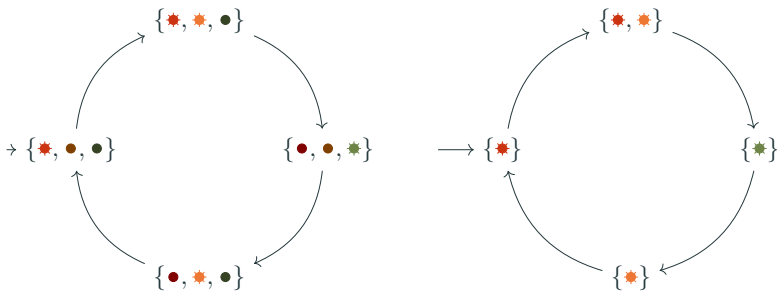
- s is not labelled with ap because ap does not hold in s .
- s is not labelled with ap because we do not know whether ap holds/should hold in s .

E.g. in developing a temporal model from an artefact we may abstract over some detail, merging states in which ap holds and in which it does not hold.

Implication and negation

One may sometimes wish to be careful about this and use implication-free (negation-free) LTL: instead making AP include for each ap also nap for the contrary of ap . Then one can distinguish:

1. $ap \in \ell(s)$
2. $nap \in \ell(s)$
3. $\{ap, nap\} \cap \ell(s) = \emptyset$



Counting steps

$G (Call_1 \rightarrow (Loc_1 \vee X (Loc_1) \vee \dots \vee X (X (X (X (X (Loc_1)))))))$

“If there is a call of the elevator to level 1, the elevator will get to level 1 in at most 5 steps.”

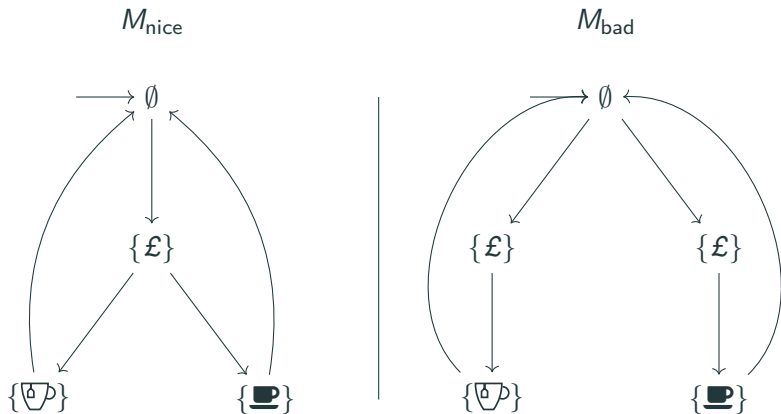
Counting steps, as in the specification above, is only useful if the temporal model corresponds well enough to the real artefact.

Branching

“The path of the elevator is not entirely idiotic.” One possibly desirable property: “If it is possible to answer a call to some level in the next step, then the elevator does that.”

LTL cannot express this: it cannot express properties relating to the different possible transitions out of a state. In the next lecture we will see CTL, which can express such properties.

Tea & coffee machines



A good property about M_{nice} : “Following payment, it is **possible** to receive coffee in the next state”. We cannot say this in LTL.

Summary

LTL formulae allow specifying temporal models, by describing the properties of infinite paths in the model. LTL has a linear notion of time, in which each state in a path has a unique successor state.