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:

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

- ✔ It works.
- ✔ It has a well-understood theory.
- ✗ It is very error-prone.
- ✗ It 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 $AP$, $\phi, \ldots \in \text{PathProp} ::=$

- $\perp$ | false
- $\top$ | true
- $\text{injp } p$ | atomic predicate
- $\neg \phi$ | negation
- $\phi_1 \land \phi_2$ | conjunction
- $\phi_1 \lor \phi_2$ | disjunction
- $\phi_1 \rightarrow \phi_2$ | implication
- $X \phi$ | $\text{neXt}$
- $G \phi$ | 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

- **X ϕ**: the tail of the current path satisfies ϕ
- **G ϕ**: every suffix of the current path satisfies ϕ
- **F ϕ**: some suffix of the current path satisfies ϕ
- **ϕ₁ U ϕ₂**: some suffix of the current path satisfies ϕ₂, and all the suffixes of the current path of which that path is a suffix satisfy ϕ₁

Semantics of LTL – Examples

Consider the path \( \pi = s₀, s₁, s₂, s₃, \ldots \). Then:

- \( \pi \) satisfies \( a \)
- \( \pi \) does not satisfy \( b \land X c \)
- \( \pi \) satisfies \( a \rightarrow F c \)
- \( \pi \) satisfies \( F (G c) \)
- \( \pi \) does not satisfy \( G (F a) \)

Semantics of LTL – Examples (2)

Consider the path \( \pi = s₀, s₁, s₂, s₃, \ldots \). Then:

- \( \pi \) does not satisfy \( G (F (a \land b \land d)) \)
- \( \pi \) satisfies \( (G (F (a)) \land G (F (b)) \land G (F (d))) \)
- \( \pi \) does not satisfy \( G ((c \land a) \rightarrow X d) \)
- \( \pi \) satisfies \( G (F (c \land X c)) \)

Notation

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

- \( □ϕ \) instead of \( G ϕ \)
- \( ⋄ϕ \) instead of \( F ϕ \)
- \( ○ϕ \) instead of \( X ϕ \)
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:

$$M, \ldots \in \text{TModel} \overset{\text{def}}{=} (S \in \text{Set}) \times \text{states} \times (S_0 \in \text{sub } S) \times \text{initial states} \times (T \not\in \text{relation } S \times S) \times \text{transition} \times (\ell \in (S \rightarrow \text{sub } AP)) \times \text{state labelling}$$

such that $T$ is left-total:

$$\forall s \in S. \exists s' \in S. s \xrightarrow{T} s'$$

Semantics of LTL

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

We define whether $M$ satisfies $\phi$:

$$M \models \phi \overset{\text{def}}{=} \forall s \in M. S. (M, S_0, s) \Rightarrow (s \models^s_M \phi)$$

$$s \models^s_M \phi \overset{\text{def}}{=} \left( \forall \pi \in \text{stream } M. S. \text{IsPath } M \pi \land \pi^0 = s \Rightarrow \pi \models^p_M \phi \right)$$
Semantics of LTL

\[ \pi \models^p X \phi \overset{\text{def}}{=} (\text{tailn } M.S 1 \pi) \models^p \phi \]

\[ \pi \models^p F \phi \overset{\text{def}}{=} \exists n \in \mathbb{N}. (\text{tailn } M.S n \pi) \models^p \phi \]

\[ \pi \models^p G \phi \overset{\text{def}}{=} \forall n \in \mathbb{N}. (\text{tailn } M.S n \pi) \models^p \phi \]

\[ \pi \models^p \phi_1 U \phi_2 \overset{\text{def}}{=} \exists n \in \mathbb{N}. \left( \forall k \in \mathbb{N}. 0 \leq k < n \Rightarrow (\text{tailn } M.S k \pi) \models^p \phi_1 \right) \land (\text{tailn } M.S n \pi) \models^p \phi_2 \]

LTL examples: Goat puzzle

Assume \( AP \) includes such atomic propositions as:

- \( \square L \): "the cabbage is on the left side of the river", and
- \( \Diamond R \): "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: 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.

A model satisfying Safe \& Live is a solution to the puzzle.
LTL examples: elevator

- “The door is not open at half-levels.”
  \[ \text{G } ((\text{Loc}_{0.5} \lor \text{Loc}_{1.5}) \rightarrow \text{Closed}) \]
  (“The door is closed when the elevator is between levels.”)
- “The elevator does not lock people in.”
  \[ \text{G } (\text{F } (\text{Open})) \]
  (“For every state along the path there is a subsequent state in
  which the door is open infinitely often.”)
  Or: \[ \neg (\text{F } (\text{G } (\text{Closed}))) \]

LTL examples. Updated wrt handout.

- “If the elevator is called to a level, then it eventually gets
  there.”
  \[ \text{G } (\text{Call}_2 \rightarrow \text{F } (\text{Loc}_2)) \]
- Maybe: \[ \text{G } (\text{Call}_2 \rightarrow (\text{Call}_2 \cup (\text{Loc}_2 \land \text{Open}))) \]
  “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 \( \text{ap} \));
does \( \text{ap} \) hold for a path \( \pi \) where \( s = \pi_0 \) is not labelled with \( \text{ap} \)?

\[ \pi \models^p \neg \text{ap} = \neg (\pi \models^p \text{ap}) = \neg (M.\ell (\pi_0) \text{ ap}) = \top \]

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

- \( s \) is not labelled with \( \text{ap} \) because \( \text{ap} \) does not hold in \( s \).
- \( s \) is not labelled with \( \text{ap} \) because we do not know whether \( \text{ap} \)
  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 \( \text{ap} \) 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 \( \text{AP} \)
include for each \( \text{ap} \) also \( \neg \text{ap} \) for the contrary of \( \text{ap} \). Then one can distinguish:

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

E.g. in developing a temporal model from an artefact we may
abstract over some detail, merging states in which \( \text{ap} \) holds
and in which it does not hold.
Counting steps

\[ G (\text{Call}_1 \rightarrow (\text{Loc}_1 \lor X \text{ (Loc}_1) \lor \ldots \lor X (X (X (X (X (\text{Loc}_1)))))))) \]

“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 \(M_{\text{nice}}\): “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.