Chapter 9

Linear Temporal Logic (LTL)

This chapter introduces the Linear Temporal Logic (LTL) to reason about state properties of Labelled Transition Systems defined in the previous chapter. We will first introduce the syntax of LTL. Then, we will give path semantics to LTL.

9.1 Syntax

A valid LTL formula has the following syntax:

\[ \varphi ::= a | \varphi_1 \land \varphi_2 | \neg \varphi | X\varphi | \varphi_1 U \varphi_2 \]

where:

- \( a \) is an atomic proposition
- \( \varphi, \varphi_1 \) and \( \varphi_2 \) are valid LTL formulas
- \( X \) denotes the "next" operator (see below)
- \( U \) denotes the "until" operator (see below)

**Precedence** The unary operators bind stronger than the binary ones. Negation (\( \neg \)) and \( X \) bind equally strong. Temporal operator until (\( U \)) takes precedence over the Boolean connective (\( \land \)). Operator \( U \) is right associative, that is, \( \varphi_1 U \varphi_2 U \varphi_3 \) stands for \( \varphi_1 (\varphi_2 U \varphi_3) \).

From negation and conjunction, we define the usual Boolean connectives using De Morgan’s rules:

- \( \varphi \lor \psi \equiv \neg (\neg \varphi \land \neg \psi) \) (disjunction)
- \( \varphi \rightarrow \psi \equiv \neg \varphi \lor \psi \) (implication)

From the syntax, we define the following:

- \( \text{True} \equiv \varphi \lor \neg \varphi \)
- \( \text{False} \equiv \neg \text{True} \)
9.2 Semantics over Paths

In this section, we define the semantics of the main operators of LTL over paths.

Atomic Proposition  An atomic proposition is true on a path, if it holds on the first state of the path. On top of Figure 9.1, proposition $a$ holds at the first position, so it holds for that path. Formally, given a path $\pi$ and atomic proposition $a$ we have the following definition:

$$\pi \models a \equiv a \in L(\pi[0])$$

Negation of atomic proposition  The negation of an atomic proposition holds on a path $\pi$ if and only if the atomic proposition does not hold in the first state of the path:

$$\pi \models \neg a \equiv a \notin L(\pi[0])$$

Conjunction  Given two LTL formulas $\varphi$ and $\psi$, the conjunction holds on a path $\pi$ if and only if both formulas hold on the path.

$$\pi \models \varphi \land \psi \equiv \pi \models \varphi \text{ and } \pi \models \psi$$

Next  The next operator is used to specify that a formula holds at next, that is, at the next position in a path. Given proposition $a$, formula $Xa$ holds on a path if it holds at the next state, that is, state at position 1 on the path. On top of Figure 9.1, $X\neg a$ holds as $a \notin L(\pi[1[..])$. In the line below, you can see a path where $Xa$ holds.
9.3. SEMANTICS OVER TRANSITION SYSTEMS

\[ \pi \models X\varphi \equiv \pi[1..] \models \varphi \]

**Until**  The until operator specifies that a formula is true until another one is true. There are two parts in the definition of \( \varphi \mathcal{U} \psi \):

1. formula \( \psi \) must hold at some position on the path;
2. at all previous positions, formula \( \varphi \) must hold.

Formally, the definition is as follows:

\[ \pi \models \varphi \mathcal{U} \psi \equiv \exists i \geq 0. \pi[i..] \models \psi \text{ and } \forall 0 \leq j < i. \pi[j..] \models \varphi \]

It is important to notice that \( \varphi \mathcal{U} \psi \) enforces \( \psi \) to hold but it does not enforce \( \varphi \) to be true. If \( \psi \) holds at all positions, \( \varphi \mathcal{U} \psi \) also holds.

From these two temporal operators, we define the semantics of two additional operators: eventually and globally.

**Eventually**  This operator is noted \( \mathcal{F} \) and sometimes \( \mathcal{U} \). Intuitively (see Figure 9.1) \( \mathcal{F}\varphi \) means that \( \varphi \) must hold somewhere in the future. Formally, \( \mathcal{F}\varphi \) is defined as \( \text{True} \mathcal{U} \varphi \). This expands to the following semantics:

\[ \pi \models \mathcal{F}\varphi \equiv \exists i \geq 0. \pi[i..] \models \varphi \]

**Globally**  This operator is noted \( \mathcal{G} \) or sometimes \( \mathcal{A} \). This is the dual of the eventually operator. Intuitively (see Figure 9.1), \( \mathcal{G}\varphi \) means that \( \varphi \) always holds. It is formally defined as \( \neg \mathcal{F}\neg\varphi \), which expands to the following semantics:

\[ \pi \models \mathcal{G}\varphi \equiv \forall i \geq 0. \pi[i..] \models \varphi \]

**9.3 Semantics over Transition Systems**

Once we have defined the semantics of the LTL operators over paths, we now define when a transition system satisfies an LTL formula. Intuitively, an LTS satisfies an LTL formula if and only if the formula holds for all paths of the LTS.

**Validity in a state**  An LTL formula \( \varphi \) is valid in state \( s \) if and only if \( \varphi \) holds for all paths starting in \( s \).

\[ s \models \varphi \equiv \forall \pi \in \text{Paths}(s). \pi \models \varphi \]

**Validity for an LTS**  An LTS formula \( \varphi \) is valid for LTS \( T \) if and only \( \varphi \) holds in all the initial states of \( T \). This means that \( \varphi \) holds in all paths starting from any initial state.

\[ T \models \varphi \equiv \forall s_0 \in I.s_0 \models \varphi \]
9.4 Example

Let’s have a look at an example. Consider the model of a simple beverage vending machine in Figure 9.2. On this machine, we can express the following requirement:

The system shall produce tea if and only if a coin has been inserted and the start button has been pressed.

This requirement states that if a coin has been inserted and the start button has been pressed, users should have tea. It also states that users should have tea then a coin has been inserted and the start button has been pressed.

To express this formula, we can define the following atomic propositions:

- \( \text{coin1} \) is true if and only if a coin has been inserted. Coins are consumed after a beverage has been produced.
- \( \text{start} \) is true if and only if the start button has been pressed. This proposition is reset when beverage has been produced.
9.5. NOTE ON NEGATION

- ServingTea is true if and only if the system is about to serve tea.

Based on this textual definition, we can define the following labelling function:

- \( L(s_1) = \{\text{coin1}\} \)
- \( L(s_2) = \{\text{coin1}, \text{start}, \text{ServingTea}\} \)
- \( L(s) = \emptyset \), otherwise.

We can then formalise our requirement as follows:

\[
G((\text{coin1} \land \text{start}) \rightarrow \text{ServingTea}) \land ((\text{ServingTea} \rightarrow (\text{coin1} \land \text{start})))
\]

9.5 Note on negation

For a path, it holds that a property is true if and only if its negation is not true, that is,

\[ \pi \models \varphi \iff \pi \not\models \lnot \varphi \]

For an LTS, this equivalence does not hold. An LTS might not satisfy a property and also not satisfy the negation of this property. Consider the LTS in Figure 9.3. This LTS does not satisfy \( Fa \) nor does it satisfy \( \lnot Fa \). Counter-examples to the first formula are paths going to the right from the initial state. Counter-examples to the second formula are paths going to the left from the initial state.

9.6 Several laws

9.6.1 Duality

- \( \lnot G \varphi \equiv F \lnot \varphi \)
• $\neg F\phi \equiv G\neg \phi$

• $\neg X\phi \equiv X\neg \phi$

### 9.6.2 Idempotency

• $GG\phi \equiv G\phi$

• $FF\phi \equiv F\phi$

• $\phi \ U (\phi \ U \psi) \equiv \phi \ U \psi$

• $(\phi \ U \psi) \ U \psi \equiv \phi \ U \psi$

### 9.6.3 Absorption

• $FG\phi \equiv GF\phi$

• $GFG\phi \equiv FG\phi$

### 9.6.4 Distributivity

• $X(\phi \ U \psi) \equiv (X\phi) \ U (X\psi)$

• $F(\phi \lor \psi) \equiv (F\phi) \lor (F\psi)$

• $G(\phi \land \psi) \equiv (G\phi) \land (G\psi)$

Note the following:

• $F(\phi \land \psi) \neq (F\phi) \land (F\psi)$

• $G(\phi \lor \psi) \neq (G\phi) \lor (G\psi)$

### 9.7 Conclusion

We have defined Linear Temporal Logic (LTL) to reason about the infinite behaviour of Labelled Transition Systems. The validity of LTL formulas is defined over paths and an LTS satisfies an LTL formula if and only if it satisfies the formula on all paths starting from any initial state. To show that an LTS violates a property, it is sufficient to exhibit a counter-example, that is, a path for which the property is not true.
9.8 Exercises

Exercise 9.8.1. Consider the formulas $\varphi = a \ U \ (b \ U \ c)$ and $\psi = (a \ U \ b) \ U \ c$. Prove that these formulas are equivalent or exhibit a counter-example, that is, a path such that one formula holds on the path and the other doesn’t.

Exercise 9.8.2. Consider a traffic light. Write an LTL formula that formalises the requirement that the light must change colours in the following sequence: red, yellow, and green.