


Temporal logics



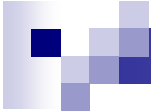
Temporal logics (TL) offers means for **analysis of dynamic** (imperative) **state systems** and for **verification of their properties**. The role of TL in this context can be compared to that of classical logic for systems of mathematics.

Important properties in formal verification that can be expressed using linear temporal logic:

- safety properties usually state that something bad never happens,
- liveness properties state that something good keeps happening.

Intuitively, the state systems are described by their behaviour during their way (travel) through sequences of such states.

Examples: program modules, communication protocols, DataBase systems, logic circuits, chips, computational processes



Recommendation for studying this text: The important parts are highlighted in yellow (similarly to this paragraph). The remaining text explains the context or offers some additional information.

We will be concerned with the propositional version of TL.

States. Let the states be described by **primitive propositions** expressing simple statements of the type „the global variable x has the value 10“.

Each state is characterized by its evaluation of primitive propositions (it can be either finite or infinite).



Time. Computation processes advance in individual steps =>
we consider discrete time.

Future of the system can be modelled by diverse ordering of
the considered time points.

The simplest (and rather realistic) choice is to consider their
linear ordering.

In this case the time points are represented by a **finite** or
infinite sequence numbered by natural numbers

$$m_0, m_1, m_2, m_3, \dots, m_n, m_{n+1}, \dots$$



This ordering is characteristic for *propositional Linear Temporal Logics (LTL)*.

In more complex cases, e.g. tree-like structures are applied. In this case we are talking about branching time.

Let V be the set of the primitive formulas. The weakest language of propositional temporal logics L_{LTL} consists of

- all formulas from V and
- symbols **false** \rightarrow \square \circ $(\)$.

This language is often enhanced by

- additional connectives, namely \vee , $\&$ (\wedge) and \equiv and
- further operators, e.g. unary op. \diamond , binary op. U („until“).



Temporal operators \circ , \square , \diamond are described by the following English expressions

- \circ *nexttime* or simply *next*,
- \square *always* or *henceforth* and
- \diamond *sometime*.

The formulas $\circ A$, $\square A$ and $\diamond A$ have the following English and (Czech) reading

- $\circ A$: *nextA* (*příště A*),
- $\square A$: *alwaysA*, (*vždy A*)
- $\diamond A$: *sometimeA*, (*někdy A*)

Preference \neg , \circ , \square , \diamond bind more strongly than the classic connectives \vee , \wedge , \rightarrow and \equiv .

Temporal operators and natural language

Let us introduce following abbreviations

- α „*The Moon circulates around the Earth.*“
- β „*The moon is rising*“,
- γ „*The moon is setting down*“.

- $\diamond\beta$, expresses the claim „*The moon will rise once*“,
- $\square\diamond\beta$ means „*The moon will be rising again and again.*“,
- $\square(\beta \rightarrow \diamond\gamma)$ means „*Whenever the moon will rise, it will set down later*“.



Definition. (LTL-formula constructed from the set V of primitive formulas)

1. Any primitive formula from the set V is a LTL-formula.
2. Propositional constant **false** is a LTL-formula.
3. If A and B are LTL-formulas, $A \rightarrow B$ is LTL-formula, too.
4. If A is a LTL-formula, then $\circ A$ and $\square A$ are LTL formulas.

Other connectives and the operator \diamond are understood as abbreviations for more complex constructs, namely:

- $\neg A$ stands for $\neg (\mathbf{false} \rightarrow A)$
- **true** stands for $\neg \mathbf{false}$
- $\diamond A$ stands for $\neg \square \neg A$
- the connectives \vee , $\&$ (\wedge), \equiv have the same meaning as in classic logic.

Informal reading for some frequently used temporal formulas.

$A \rightarrow \circ B$ „If A holds now, B will be true in the following (next) state.“

$A \rightarrow \square B$ „If A holds now, then B is true now as well and it will remain true in all the states from now on“

$A \rightarrow \diamond B$ „If A holds now, B is true now or in at least one of the states that will follow in future.“

$\square(A \rightarrow B)$ „For all the states from now on there holds that if A is true then B must be true as well“

$\square \diamond A$ „Now and for all the future states there holds that A is true now or in at least one of the states that will follow in future.“

„(Since now) A will be true in infinitely many states.“

„The formula A will be true again and again.“

$\diamond \square A$ „There will occur a state in which A becomes true and it will remain so.“

„ A is true almost ever (from certain instant on).“

Semantics of Linear Temporal Logics (LTL)

Let V be the set of elementary formulas.

Temporal (or Kripke) *structure* for V is an infinite sequence $\mathbf{M} = (\eta_0, \eta_1, \eta_2, \dots)$ of evaluations

$$\eta_i : V \Rightarrow \{ff, tt\}$$

providing truth values (tt for true and ff for false) to all the primitive formulas. Each evaluation is referred to as a *state* (*world*).

The state η_0 is *the initial state* of \mathbf{M} in the time point m_0 and η_{n+1} is the state following the state η_n .

Definition.

Let us consider the formula A , the temporal structure M and the index i . The **truth value** $M_i(A)$ of A *in the time point* i of M is defined **by induction**:

$$M_i(p) = \eta_i(p) \quad p \in V$$

$$M_i(\text{false}) = \text{ff}$$

$$M_i(A \rightarrow B) = \text{tt} \quad \text{iff}^* \quad M_i(A) = \text{ff} \quad \text{or} \quad M_i(B) = \text{tt}$$

For the operators

$$M_i(\circ A) = M_{i+1}(A)$$

$$M_i(\Box A) = \text{tt} \quad \text{iff} \quad M_j(A) = \text{tt} \quad \text{for every } j \geq i$$

*

iff stands for *if and only if* and it is sometimes expressed by the symbol \Leftrightarrow

Definition continued for the defined symbols

- $M_i(\neg A) = \text{tt}$ *iff* $M_i(A) = \text{ff}$
- $M_i(A \vee B) = \text{tt}$ *iff* $M_i(A) = \text{tt}$ *or* $M_i(B) = \text{tt}$
- $M_i(A \& B) = \text{tt}$ *iff* $M_i(A) = \text{tt}$ *and* $M_i(B) = \text{tt}$
- $M_i(A \equiv B) = \text{tt}$ *iff* $M_i(A) = M_i(B)$
- $M_i(\diamond A) = \text{tt}$ *iff* $M_j(A) = \text{tt}$ for some $j \geq i$

Interesting observation:

$$\begin{aligned} M_i(\diamond A) = \text{tt} & \quad \textit{iff} \quad M_i(\neg \square \neg A) = \text{tt} \\ & \quad \textit{iff} \quad M_i(\square \neg A) = \text{ff} \\ & \quad \textit{iff} \quad M_j(\neg A) = \text{ff} \quad \textit{for some } j \geq i \\ & \quad \textit{iff} \quad M_j(A) = \text{tt} \quad \textit{for some } j \geq i \end{aligned}$$

Definition (validity, semantic or logical consequence)

Let A be a LTL-formula with a set of primitive variables V and let T be a set of formulas of the same language.

The formula A is *valid in a temporal structure* M for V (or A is *true in* M), if $M_i(A) = \text{tt}$ for all i . This is denoted as $M \models A$ or $\models_M A$.

The structure M is a model of the set of formulas T , if $M \models B$ for all formulas B from T .

The formula A is a *semantic consequence of* T (denoted as $T \models A$) if A is valid in any model M of the set of formulas T .

A is *valid* (denoted as $\models A$), if A is valid in any temporal structure M . In other words, A is valid iff $\emptyset \models A$.

Example. $\neg \circ A \equiv \circ \neg A$ is a valid formula .

It is necessary to show, that $M_i(\neg \circ A) = M_i(\circ \neg A)$ holds in any structure M and for all its time points i .

$$\begin{aligned} M_i(\neg \circ A) = \text{tt} &\Leftrightarrow M_i(\circ A) = \text{ff} \\ &\Leftrightarrow M_{i+1}(A) = \text{ff} \\ &\Leftrightarrow M_{i+1}(\neg A) = \text{tt} \\ &\Leftrightarrow M_i(\circ \neg A) = \text{tt} \end{aligned}$$



Lemma 1. (correctness of the rule Modus Ponens)

Let M be a temporal structure and $i \in \mathbb{N}$ (*the set of natural numbers*). If $M_i(A) = \text{tt}$ and $M_i(A \rightarrow B) = \text{tt}$, then $M_i(B) = \text{tt}$.

Proof. $M_i(A \rightarrow B) = \text{tt}$ iff $M_i(A) = \text{ff}$ or $M_i(B) = \text{tt}$.

Combining this with the assumption $M_i(A) = \text{tt}$, we get $M_i(B) = \text{tt}$.

Theorem 2. If $\mathbf{T} \models A$ and $\mathbf{T} \models A \rightarrow B$, then $\mathbf{T} \models B$.

Proof. Let the structure M be a model of \mathbf{T} . This means that for any i there holds both

$$M_i(A) = \text{tt} \text{ and } M_i(A \rightarrow B) = \text{tt}$$

According to Lemma 1 we can conclude that $M_i(B) = \text{tt}$ and thus $\mathbf{T} \models B$.

Theorem 3. If $\mathbf{T} \models A$, then $\mathbf{T} \models \circ A$ and $\mathbf{T} \models \square A$.
Namely there holds $A \models \circ A$ and $A \models \square A$.

Proof. Let M be any temporal lin. structure, that is a model of \mathbf{T} . We have to show that for an arbitrary natural number i we select there holds $M_i(\circ A) = \text{tt}$ and $M_i(\square A) = \text{tt}$.

According to the assumption $M_j(A) = \text{tt}$ for all $j \in \mathbb{N}$, we can be sure that this is true for the special cases bellow

$$M_{i+1}(A) = \text{tt} \quad \text{and} \quad M_k(A) = \text{tt} \quad \text{for all } k, k \geq i.$$

Thos means that

$$\mathbf{T} \not\models \circ A \quad \text{and} \quad \mathbf{T} \not\models \square A.$$

Theorem 4. If $\mathbf{T} \models A \rightarrow B$ and $\mathbf{T} \models A \rightarrow \circ A$, then $\mathbf{T} \models A \rightarrow \square B$ (the induction rule).

Proof. Let \mathbf{M} be a model of \mathbf{T} . let us distinguish 2 cases:

- If $\mathbf{M} \models \neg A$, then the claim of the theorem results from the properties of the boolean connective „ \rightarrow “.
- Suppose that $\mathbf{M} \models \neg A$ does not hold. There must exist a state k such that $\mathbf{M}_k \models A$. Let us choose the lowest i such that $\mathbf{M}_i \models A$. Since it must be the case that $\mathbf{M}_i \models A \rightarrow \circ A$, we know that $\mathbf{M}_i \models \circ A$. By induction we get $\mathbf{M}_i \models \square A$ and $\mathbf{M}_i \models \square B$ as simple consequences of introductory assumptions. Thus for all j there holds $\mathbf{M}_j \models A \rightarrow \square B$

Notation. Let $\mathbf{M} = (\eta_0, \eta_1, \eta_2, \dots)$ be some linear temporal structure for the primitive formulas V . Let i be any firmly selected natural number. Let us define the temporal structure \mathbf{M}^i , that is obtained from \mathbf{M} by “shifting time” by i steps into future:

$$\mathbf{M}^i = (\eta_0', \eta_1', \eta_2', \dots)$$

where $\eta_j' = \eta_{i+j}$ for any j . \mathbf{M}^i is again a temporal structure according to the original definition. We can describe it in more direct way as

$$\mathbf{M}^i = (\eta_i, \eta_{i+1}, \eta_{i+2}, \dots, \eta_{i+j}, \eta_{i+j+1}, \dots)$$

Theorem 6 (deduction theorem for LTL). $\mathbf{T} \cup \{A\} \models B$ právě když $\mathbf{T} \models \Box A \rightarrow B$.

Semantic proof: \Leftarrow This is clear.

\Rightarrow Suppose $\mathbf{T} \cup \{A\} \models B$ holds. Let S be a structure for \mathbf{T} such that it is not a structure for $\mathbf{T} \cup \{A\}$. This means that the set $\mathcal{L} = \{s_i \text{ from } S : S_i(A) = \text{ff}\}$ is not empty! If \mathcal{L} is finite, let us denote k its maximum number. Then

- a) for all $i < k+1$ there does not hold $\mathcal{L}_i \models \Box A$ (and thus $\mathcal{L}_i \models \Box A \rightarrow B$)
- b) for all $j > k$ there holds $S_j \models \Box A \ \& \ B$ (and thus $S_j \models \Box A \rightarrow B$).

If \mathcal{L} is infinite it is clear that $S \models \neg \Box A$ and thus $S \models \Box A \rightarrow B$

Theorem 7. If $\mathbf{T} \models A \rightarrow B$ then $\mathbf{T} \cup \{A\} \models B$.

The inverse implication does not hold in LTL! It is enough to show a counter example. Let \mathbf{T} be an empty set. By Theorem 3 there holds $A \models \Box A$ for any formula. But the implication $A \rightarrow \Box A$ does not have to be valid (consider LTL structure M such that for some i and $j > i$ there holds $M_i(A) = \text{tt}$ a $M_j(A) = \text{ff}$)

Important valid formulas

$$\models \Box \neg A \equiv \neg \Diamond A$$

$$\models \Box A \rightarrow \circ A$$

$$(1) \models \circ A \rightarrow \Diamond A$$

$$\models \Box (A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B)$$

$$\models (\Diamond \Diamond A \rightarrow \Diamond A)$$

$$\models \Box \Diamond \neg A \equiv \neg \Diamond \Box A$$

$$\models \Diamond \Box \neg A \equiv \neg \Box \Diamond A$$

$$\models \circ \neg A \equiv \neg \circ A$$

$$\models \circ (A \rightarrow B) \equiv (\circ A \rightarrow \circ B)$$

$$\models \Box A \equiv (A \ \& \ \circ \Box A)$$

Following valid implications cannot be strengthened into equivalences

$$\models A \rightarrow \Diamond A$$

$$\models \Box A \rightarrow A$$

$$\models \circ A \rightarrow \Diamond A$$

$$\models \Box A \rightarrow \circ A$$

$$\models \Box A \rightarrow \Diamond A$$

$$\models \Box A \rightarrow \circ \Box A$$

$$\models A \cup B \rightarrow \Diamond A$$

$$\models \Diamond \Box A \rightarrow \Box \Diamond A$$

Idempotency of \diamond , \square , $\square\diamond$ a $\diamond\square$

$$\models \diamond\diamond A \equiv \diamond A$$

$$\models \square\square A \equiv \square A$$

$$\models \diamond\square \diamond\square A \equiv \diamond\square A$$

$$\not\models \square\diamond \square\diamond A \equiv \square\diamond A$$

Of course, this cannot be true about the “next” operator:
the formula $\circ\circ A \equiv \circ A$ is not valid !


Combination of modalities $\square\diamond$ **and** $\diamond\square$: „consume“ all other modalities with one argument. This can be expressed as follows

$$\models (\square\diamond) A \equiv \circ (\square\diamond) A \equiv \diamond (\square\diamond) A \equiv \square (\square\diamond) A$$

$$\models \square\diamond (\square\diamond) A \equiv \diamond\square (\square\diamond) A$$

$$\models (\diamond\square) A \equiv \circ (\diamond\square) A \equiv \diamond (\diamond\square) A \equiv \square (\diamond\square) A$$

$$\models \square\diamond (\diamond\square) A \equiv \diamond\square (\diamond\square) A$$



According to their definition the operators \diamond and $\square\diamond$ describe *existential properties* while the operators \square and $\diamond\square$ can be understood as descriptors of *universal properties*.

It is no surprise that they exhibit similar behaviour as the *existential* and *universal* quantifier of propositional logic.

$$\models \diamond (A \vee B) \equiv (\diamond A \vee \diamond B) \quad \models \square\diamond (A \vee B) \equiv (\square\diamond A \vee \square\diamond B)$$

$$\models \square (A \wedge B) \equiv (\square A \wedge \square B) \quad \models \diamond\square (A \wedge B) \equiv (\diamond\square A \wedge \diamond\square B)$$



We have focused our attention to the unary operators, only. Just for completeness let us introduce interpretation for the operator “until”

Definition. $A \cup B$

For a temporal structure M , index i and the formulas A, B the meaning of the formula $M_i(A \cup B)$ is defined as follows

$$M_i(A \cup B) = \text{tt} \Leftrightarrow$$

there is a $j, j > i$ such that $M_j(B) = \text{tt}$ and for all $k, i < k < j$ $M_k(A) = \text{tt}$

The basic valid transformations for the operator \cup (*until*) and the boolean connectives (**distributivity**):

$$\models ((A \wedge B) \cup C) \equiv ((A \cup C) \wedge (B \cup C))$$

$$\models (A \cup (B \vee C)) \equiv ((A \cup C) \vee (B \cup C))$$

The operator \circ (next) refers to a single time point. That is why it can be distributed over all boolean connectives.

$$\begin{aligned} \models \circ(A \vee B) &\equiv (\circ A \vee \circ B) & \models \circ(A \wedge B) &\equiv (\circ A \wedge \circ B) \\ \models \circ(A \rightarrow B) &\equiv (\circ A \rightarrow \circ B) & \models \circ(A \equiv B) &\equiv (\circ A \equiv \circ B) \end{aligned}$$

The equivalence $\models \circ\neg A \equiv \neg \circ A$ has been mentioned earlier already.

The following formulas are valid but they cannot be strengthened into equivalences.

$$\begin{aligned} \models (\Box A \vee \Box B) &\rightarrow \Box (A \vee B) & \models \Diamond \Box (A \vee B) &\rightarrow (\Diamond \Box A \vee \Diamond \Box B) \\ \models \Diamond (A \wedge B) &\rightarrow (\Diamond A \wedge \Diamond B) & \models \Box \Diamond (A \wedge B) &\rightarrow (\Box \Diamond A \wedge \Box \Diamond B) \end{aligned}$$

The introduced operators exhibit **monotonicity** in all arguments.

$$\begin{aligned} & \models \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B) & \models \Box(A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B) \\ & \models \circ(A \rightarrow B) \rightarrow (\circ A \rightarrow \circ B) \\ & \models \Box(A \rightarrow B) \rightarrow (\Diamond \Box A \rightarrow \Diamond \Box B) & \models \Box(A \rightarrow B) \rightarrow (\Diamond \Box A \rightarrow \Diamond \Box B) \end{aligned}$$

An interesting observation: All over $\Box A \rightarrow (A \rightarrow \circ \Box A)$ is a valid formula, the formule describing the inverse implication, namely $(A \rightarrow \circ \Box A) \rightarrow \Box A$, is not valid !

The temporal operators can be characterized as fixed points.

$$\models \Diamond A \equiv (A \vee \circ \Diamond A) \qquad \models \Box A \equiv (A \wedge \circ \Box A)$$

Some characteristic relations for the “until” operator:

$$\begin{aligned} & \models \Box(A \rightarrow B) \rightarrow ((A \cup C) \rightarrow (B \cup C)) \\ & \models \Box(A \rightarrow B) \rightarrow ((C \cup A) \rightarrow (C \cup B)) \\ & \models (A \cup B) \equiv B \vee (A \wedge \circ(A \cup B)) \end{aligned}$$



Recommended reading for more detailed treatment:

- Huth M., Ryan M.: *Logic in Computer Science*, Cambridge University Press, 2004
- Manna/Pnueli: *The Temporal Verification of Reactive Systems: Progress*, Springer Verlag 1995, <http://theory.stanford.edu/~zm/tvors3.html>
- Program system **SPIN**