

---

# Chapter 0

# Introduction

# Temporal Logic

Usual (mathematical) logic (classical predicate logic)

- Provides a formal language for the precise formulation of mathematical statements, called **formulas**,
- investigates instruments to verify statements.

Temporal Logic (TL) **addresses additionally**

- reflection of time-dependent statements,
- provision of a language for formulation of temporal relationships of statements and investigation of means to verify such relationships.

## State systems

Collective (informal) notion for all kinds of systems that **execute**.

An execution is a sequence of states. Communication protocols, database systems, sequential logic systems, automata, etc. are state systems.

Application of temporal logic:

Interpreting states as “time points” allows formulations of typical statements about system executions and their (temporal) logic handling.

## Content of course:

1. Presentation of the **foundations of temporal logic(s)**
2. “Temporal logic specifications” of **state systems**  
(formal description of systems in TL)
3. Application to the **verification of state systems**  
(especially parallel programs and finite state systems)

## Literature

- Fred Kröger: Temporal Logic of Programs Springer, 1987
- Manna/Pnueli: The Temporal Logic of Reactive and Concurrent Systems -  
Specification (Vol. 1) Springer, 1992
- The Temporal Logic of Reactive and Concurrent Systems -  
Safety Properties (Vol. 2) Springer, 1992
- Gabbay/Hodkinson/Reynolds:  
Temporal Logic - Mathematical Foundations and  
Computational Aspects (Vol. 1)  
Oxford, Clarendon Press, 1994
- Colin Stirling: Modal and Temporal Properties of Processes Springer, 2001

---

# Chapter 1

## Classical Logic (Overview)

## Language of propositional logic

### Definition 1 ( $\mathcal{L}_{PL}(\mathcal{V})$ )

Let  $\mathcal{V}$  be a (finite or denumerable) set of **atomic propositions**.

A language  $\mathcal{L}_{PL}(\mathcal{V})$  (briefly:  $\mathcal{L}_{PL}$ ) of propositional logic is defined as follows:

#### *Alphabet*

- All atomic propositions of  $\mathcal{V}$ ,
- the symbols: **false** |  $\rightarrow$  |  $($  |  $)$ .

#### *Inductive definition of formulas*

1. Every *atomic proposition* of  $\mathcal{V}$  is a formula.
2. **false** is a formula.
3. If  $A$  and  $B$  are formulas then  $(A \rightarrow B)$  is a formula.

## Language continued

Abbreviations:

$$\neg A \equiv A \rightarrow \mathbf{false},$$

$$A \vee B \equiv \neg A \rightarrow B,$$

$$A \wedge B \equiv \neg(A \rightarrow \neg B),$$

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A),$$

$$\mathbf{true} \equiv \neg \mathbf{false}.$$

## Semantics

Given distinct **truth values** ff (false) and tt (true).

A valuation  $\mathbf{B}$  for a set  $\mathcal{V}$  of atomic propositions is a mapping

$$\mathbf{B} : \mathcal{V} \rightarrow \{\text{ff}, \text{tt}\}.$$

Every  $\mathbf{B}$  can be inductively extended to the set of all formulas of  $\mathcal{L}_{PL}(\mathcal{V})$ :

1.  $\mathbf{B}(v)$  for  $v \in \mathcal{V}$  is given.
2.  $\mathbf{B}(\mathbf{false}) = \text{ff}$ .
3.  $\mathbf{B}(A \rightarrow B) = \text{tt} \Leftrightarrow \mathbf{B}(A) = \text{ff} \text{ or } \mathbf{B}(B) = \text{tt}$ .

## Semantics continued

This defines  $\mathbf{B}$  for the formula abbreviations above:

$$4. \mathbf{B}(\neg A) = \text{tt} \Leftrightarrow \mathbf{B}(A) = \text{ff}.$$

$$5. \mathbf{B}(A \vee B) = \text{tt} \Leftrightarrow \mathbf{B}(A) = \text{tt} \text{ or } \mathbf{B}(B) = \text{tt}.$$

$$6. \mathbf{B}(A \wedge B) = \text{tt} \Leftrightarrow \mathbf{B}(A) = \text{tt} \text{ and } \mathbf{B}(B) = \text{tt}.$$

$$7. \mathbf{B}(A \leftrightarrow B) = \text{tt} \Leftrightarrow \mathbf{B}(A) = \mathbf{B}(B).$$

$$8. \mathbf{B}(\text{true}) = \text{tt}.$$

## Validity, Tautology

### Definition 2 (Validity, Tautology)

Formula  $A$  of  $\mathcal{L}_{PL}$  is called **valid in  $\mathbf{B}$**  ( $\models_{\mathbf{B}} A$ ) if  $\mathbf{B}(A) = \text{tt}$ .

$A$  is called **valid** or **tautology** ( $\models A$ ) if  $\models_{\mathbf{B}} A$  holds for every  $\mathbf{B}$ .

$A$  is a **consequence of** a set  $\mathcal{F}$  of formulas ( $\mathcal{F} \models A$  or  $B_1, \dots, B_n \models A$  if  $\mathcal{F} = \{B_1, \dots, B_n\}$ ,  $n \geq 1$ ) if  $\models_{\mathbf{B}} A$  holds for every  $\mathbf{B}$  with  $\models_{\mathbf{B}} B$  for all  $B \in \mathcal{F}$ .

*Note:*  $\models A \Leftrightarrow \emptyset \models A$ .

Blas

see also exercises 1 and 2

# Formal Systems

## Definition 3 (Formal Systems)

A **formal system**  $\Sigma$  consists of

- *axioms*
- a set of (derivation) *rules* of the form  $A_1, \dots, A_n \vdash B$  ( $n \geq 1$ ).

$A_1, \dots, A_n$ : **premises**,  $B$ : **conclusion**

(*Axioms,  $A_1, \dots, A_n, B$ : formulas of a language.*)

The **derivability** of a formula  $A$  in  $\Sigma$  ( $\vdash_{\Sigma} A$  or  $\vdash A$ ) is defined inductively:

1. Every axiom is derivable.
2. If the premises of a rule are derivable then the conclusion is derivable.

## Formal Systems continued

A formula  $A$  is called **derivable from** a set  $\mathcal{F}$  of formulas ( $\mathcal{F} \stackrel{\Sigma}{\vdash} A$ ,  $\mathcal{F} \vdash A$ ) if  $A$  is derivable in the formal system which results from  $\Sigma$  by taking all formulas of  $\mathcal{F}$  as additional axioms.

This implies that:

$$\vdash A \Leftrightarrow \emptyset \vdash A .$$

If  $A$  is derivable from some  $A_1, \dots, A_n$  then  $A_1, \dots, A_n \vdash A$  can be used as a **derived rule** in other derivations.

## Formal Systems continued

### Example 1 (A formal system $\Sigma_{\text{PL}}$ for propositional logic)

#### *Axioms*

- $A \rightarrow (B \rightarrow A)$ ,
- $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$ ,
- $((A \rightarrow \mathbf{false}) \rightarrow \mathbf{false}) \rightarrow A$ .

#### *Rule*

- $A, A \rightarrow B \vdash B$  (*modus ponens*).

## Formal Systems continued

Properties of  $\Sigma_{\text{PL}}$ :

Deduction theorem (and its inverse):

$$\mathcal{F} \cup \{A\} \vdash B \Leftrightarrow \mathcal{F} \vdash A \rightarrow B.$$

see also exercises 3 and 5

Soundness of  $\Sigma_{\text{PL}}$ :

$$\mathcal{F} \vdash_{\Sigma_{\text{PL}}} A \Rightarrow \mathcal{F} \models A.$$

Completeness of  $\Sigma_{\text{PL}}$ :

$$\mathcal{F} \models A \Rightarrow \mathcal{F} \vdash_{\Sigma_{\text{PL}}} A.$$

Soundness and completeness imply:

$$\vdash_{\Sigma_{\text{PL}}} A \Leftrightarrow \models A.$$

# First-Order Logic

First-Order Logic in the glossary

## Definition 4 (Signature)

A **signature**  $SIG = (\mathbf{S}, \mathbf{F}, \mathbf{P})$  is given by

- a set  $\mathbf{S}$  of *sorts*,
- for every  $\sigma \in \mathbf{S}^*$  and  $s \in \mathbf{S}$  a (finite or denumerable) set  $\mathbf{F}^{(\sigma,s)}$  of *function symbols* (also called *constants* in the case of  $\sigma = \varepsilon$ ) and  $\mathbf{F} = \bigcup_{\sigma \in \mathbf{S}^*, s \in \mathbf{S}} \mathbf{F}^{(\sigma,s)}$ ,
- for every  $\sigma \in \mathbf{S}^*$  a (finite or denumerable) set  $\mathbf{P}^{(\sigma)}$  of *predicate symbols* (also called *atomic propositions* in the case of  $\sigma = \varepsilon$ ) and  $\mathbf{P} = \bigcup_{\sigma \in \mathbf{S}^*} \mathbf{P}^{(\sigma)}$ .

**Note:**  $f^{(\sigma,s)}$  abbreviates  $f \in \mathbf{F}^{(\sigma,s)}$ , analogously for  $p \in \mathbf{P}$ .

# First-Order Language

## Definition 5 (First-Order Language)

Given a signature  $SIG = (\mathbf{S}, \mathbf{F}, \mathbf{P})$ .

**First-order language**  $\mathcal{L}_{\text{FOL}}(SIG)$  (briefly:  $\mathcal{L}_{\text{FOL}}$ ):

### *Alphabet*

- all symbols of  $\mathbf{F}$  and  $\mathbf{P}$
- for every  $s \in \mathbf{S}$  denumerably many (*individual*) *variables*  
( $\mathcal{X}_s$  denotes the set of variables for  $s \in \mathbf{S}$ ;  $\mathcal{X} = \bigcup_{s \in \mathbf{S}} \mathcal{X}_s$ )
- the *equality symbol*  $=$
- the symbols **false** |  $\rightarrow$  |  $\exists$  | , | ( | )

## First-Order Language continued

### *Inductive definition of terms and their sorts*

1. Every variable  $x \in \mathcal{X}_s$  is a term of sort  $s$ .
2. If  $f \in \mathbf{F}^{(s_1 \dots s_n, s)}$  is a function symbol and  $t_i$  are terms of sorts  $s_i$  for  $1 \leq i \leq n$  then  $f(t_1, \dots, t_n)$  is a term of sort  $s$ .

An *atomic formula* is a string of the form

- $p(t_1, \dots, t_n)$ , where  $p \in \mathbf{P}^{(s_1 \dots s_n)}$  is a predicate symbol and  $t_i$  are terms of sorts  $s_i$  for  $1 \leq i \leq n$ , or
- $t_1 = t_2$ , where  $t_1$  and  $t_2$  are terms of the same sort.

## First-Order Language continued

### *Inductive definition of formulae*

1. *Every atomic formula is a formula.*
2. **false** is a formula, and if  $A$  and  $B$  are formulas then  $(A \rightarrow B)$  is a formula.
3. If  $A$  is a formula and  $x$  is a variable then  $\exists xA$  is a formula.

Abbreviations (additionally to  $\mathcal{L}_{\text{PL}}$ ):

$$\forall xA \equiv \neg \exists x \neg A,$$

$$t_1 \neq t_2 \equiv \neg t_1 = t_2.$$

**Note:** Write  $f$  instead of  $f()$  for constants  $f \in \mathbf{F}^{(\varepsilon, s)}$

and  $p$  instead of  $p()$  for atomic propositions  $p \in \mathbf{P}^{(\varepsilon)}$

## First-Order Language continued

### Definition 6 (Bound and Free Variables)

A variable  $x$  (more precisely: an occurrence of  $x$ ) in a formula  $A$  is called

- **bound** if it appears in some part  $\exists xB$  of  $A$ ;
- otherwise it is called **free**

### Notation:

$A_x(t)$  denotes the result of substituting  $t$  for every free occurrence of  $x$  in  $A$ .

( $x, t$  of the same sort,  $t$  without variables that are bound in  $A$ )

## Semantics

A structure  $S$  for the signature  $SIG = (\mathbf{S}, \mathbf{F}, \mathbf{P})$  consists of

- a non-empty set  $|S|_s$  (called **universe**) for every  $s \in \mathbf{S}$  and  $|S| = \bigcup_{s \in \mathbf{S}} |S|_s$ ,
- a mapping  $S(f) : |S|_{s_1} \times \dots \times |S|_{s_n} \rightarrow |S|_s$  for every function symbol  $f \in \mathbf{F}^{(s_1 \dots s_n, s)}$ ,
- a mapping  $S(p) : |S|_{s_1} \times \dots \times |S|_{s_n} \rightarrow \{\text{ff}, \text{tt}\}$  for every predicate symbol  $p \in \mathbf{P}^{(s_1 \dots s_n)}$ .

## Valuation of Formulas

A **variable valuation**  $\xi$  (with respect to  $S$ ) assigns some  $\xi(x) \in |S|_s$  to every variable  $x \in \mathcal{X}_s$  (for all  $s \in S$ ).

Inductive definition of  $S^{(\xi)}(t) \in |S|$  (for every term  $t$ ):

1.  $S^{(\xi)}(x) = \xi(x)$  for  $x \in \mathcal{X}$ .
2.  $S^{(\xi)}(f(t_1, \dots, t_n)) = S(f)(S^{(\xi)}(t_1), \dots, S^{(\xi)}(t_n))$ .

Definition of  $S^{(\xi)}(A) \in \{\text{ff}, \text{tt}\}$  for every atomic formula:

1.  $S^{(\xi)}(p(t_1, \dots, t_n)) = S(p)(S^{(\xi)}(t_1), \dots, S^{(\xi)}(t_n))$ .
2.  $S^{(\xi)}(t_1 = t_2) = \text{tt} \Leftrightarrow S^{(\xi)}(t_1)$  and  $S^{(\xi)}(t_2)$  are equal values in  $|S|_s$   
(where  $s$  is the sort of  $t_1$  and  $t_2$ ).

## Valuation of Formulas continued

Inductive extension of  $S^{(\xi)}(A)$  to all formulas:

1.  $S^{(\xi)}(A)$  for atomic formulas is already defined.
2.  $S^{(\xi)}(\mathbf{false}) = \text{ff}$ .
3.  $S^{(\xi)}(A \rightarrow B) = \text{tt} \Leftrightarrow S^{(\xi)}(A) = \text{ff} \text{ or } S^{(\xi)}(B) = \text{tt}$ .
4.  $S^{(\xi)}(\exists xA) = \text{tt} \Leftrightarrow$  there is a  $\xi'$  with  $\xi \sim_x \xi'$  and  $S^{(\xi')}(A) = \text{tt}$ .  
(Where  $\xi \sim_x \xi' \Leftrightarrow \xi(y) = \xi'(y)$  for all  $y \in \mathcal{X}$  other than  $x$ )

For  $\forall xA$  we obtain:

5.  $S^{(\xi)}(\forall xA) = \text{tt} \Leftrightarrow S^{(\xi')}(A) = \text{tt}$  for all  $\xi'$  with  $\xi \sim_x \xi'$ .

## Validity and Consequence

### Definition 7 (Validity and Consequence)

A formula  $A$  of  $\mathcal{L}_{\text{FOL}}$  is called *valid in  $S$*  (or  $S$  *satisfies*  $A$ ) ( $\models_S A$ ) if  $S^{(\xi)}(A) = \text{tt}$  for every variable valuation  $\xi$ .

$A$  is called *valid* (denoted by  $\models A$ ) if  $\models_S A$  holds for every  $S$ .

$A$  is a *consequence* of a set  $\mathcal{F}$  of formulas ( $\mathcal{F} \models A$  or  $B_1, \dots, B_n \models A$ ) if  $\models_S A$  holds for every  $S$  with  $\models_S B$  for all  $B \in \mathcal{F}$ .

(We have again  $\models A \Leftrightarrow \emptyset \models A$ .)

## A Formal System $\Sigma_{\text{FOL}}$

### Example 2 (A formal system $\Sigma_{\text{FOL}}$ for first-order logic)

A sound and complete axiomatization of FOL (formulated independently of the concrete language  $\mathcal{L}_{\text{FOL}}(\text{SIG})$ ):

#### Axioms

- all axioms of  $\Sigma_{\text{PL}}$ ,
- $A_x(t) \rightarrow \exists xA$ ,
- $x = x$ ,
- $x = y \rightarrow (A \rightarrow A_x(y))$ .

see also exercise 4

#### Rules

- $A, A \rightarrow B \vdash B$ ,
- $A \rightarrow B \vdash \exists xA \rightarrow B$  if there is no free occurrence of  $x$  in  $B$  (*particularization*).

## (First-Order) Theories

### Definition 8 ((First-Order) Theory)

A **(first-order) theory**  $Th = (\mathcal{L}_{\text{FOL}}(SIG), \mathcal{A})$  is given by

- a language  $\mathcal{L}_{\text{FOL}}(SIG)$
- a set  $\mathcal{A}$  of formulas of  $\mathcal{L}_{\text{FOL}}(SIG)$  (*non-logical axioms*)

A structure  $S$  for  $SIG$  satisfying all formulas of  $\mathcal{A}$  is called a **model** of the theory  $Th$ . Given a class  $\mathcal{C}$  of structures for a signature  $SIG$ , a  **$\mathcal{C}$ -theory** is a theory  $Th$  such that all structures of  $\mathcal{C}$  are models of  $Th$ .

A formula  $F$  of  $\mathcal{L}_{\text{FOL}}(SIG)$  is valid in all structures of  $\mathcal{C}$  if  $\mathcal{A} \models_{\Sigma_{\text{FOL}}} F$ .

see also exercise 6

### Example 3 (Group Theory)

$$SIG_{gr} = (\{G\}, \{e^{(\varepsilon, G)}, \circ^{(GG, G)}, I^{(G, G)}\}, \emptyset)$$

$Group = (\mathcal{L}_{FOL}(SIG_{gr}), \mathcal{G})$  with  $\mathcal{G}$  consisting of the formulas

$$(x \circ y) \circ z = x \circ (y \circ z)$$

$$e \circ x = x$$

$$I(x) \circ x = e$$

*Formulas  $F$  valid in all in groups can be obtained by derivations*

$$\mathcal{G} \vdash_{\Sigma_{FOL}} F$$

**Note:**  $x \circ y$  instead of  $\circ(x, y)$

## Example 4 (Natural Numbers (Peano Arithmetic))

$$SIG_{Nat} = (\{NAT\}, \mathbf{F}, \emptyset)$$

$$\text{where } \mathbf{F} = \{0^{(\varepsilon, NAT)}, s^{(NAT, NAT)}, +^{(NAT NAT, NAT)}, *^{(NAT NAT, NAT)}\}$$

$Nat = (\mathcal{L}_{FOL}(SIG_{Nat}), \mathcal{N})$  with  $\mathcal{N}$  consisting of the formulas

$$s(x) \neq 0,$$

$$s(x) = s(y) \rightarrow x = y,$$

$$x + 0 = x,$$

$$x + s(y) = s(x + y),$$

$$x * 0 = 0,$$

$$x * s(y) = (x * y) + x,$$

$$(A_x(0) \wedge \forall x(A \rightarrow A_x(s(x)))) \rightarrow \forall xA.$$

## Example 5 (Stacks)

$$SIG_{st} = (\{OBJ, STACK\}, \mathbf{F}, \emptyset)$$

where

$$\mathbf{F} = \{EMPTY(\varepsilon, STACK), PUSH(STACK OBJ, STACK), \\ POP(STACK, STACK), TOP(STACK, OBJ)\}$$

$$Stack = (\mathcal{L}_{\text{FOL}}(SIG_{st}), \mathcal{S})$$

$$\mathcal{S}: \quad PUSH(s, x) \neq EMPTY,$$

$$POP(PUSH(s, x)) = s,$$

$$TOP(PUSH(s, x)) = x$$

(where  $x \in \mathcal{X}_{STACK}, s \in \mathcal{X}_{OBJ}$ )

---

# Chapter 2

## Basic Propositional Linear Temporal Logic

# Basic Propositional Linear Temporal Logic

Goal: Assume given atomic formulas  $A, B, \dots$ :

Formulate new propositions about  $A, B, \dots$  to hold at points of time in future.

Basic operators:

$\bigcirc A$ : “A holds at the next point of time” (**nexttime** operator)

$\square A$ : “A holds at all coming points of time” (**always/henceforth** operator)

$\diamond A$ : “A holds at some point of time” (**sometime/eventually** operator)

## Basic Language of $\mathcal{L}_{\text{LTL}}(\mathcal{V})$

### Definition 9 (Basic Language of $\mathcal{L}_{\text{LTL}}(\mathcal{V})$ )

Let  $\mathcal{V}$  be a finite or denumerable set of *atomic propositions*.

*Alphabet* of  $\mathcal{L}_{\text{LTL}}(\mathcal{V})$  (briefly:  $\mathcal{L}_{\text{LTL}}$ ):

- all *atomic propositions* of  $\mathcal{V}$ ,
- the symbols **false** |  $\rightarrow$  |  $\circ$  |  $\square$  |  $($  |  $)$ .

Inductive definition of **formulas**:

1. Every *atomic proposition* of  $\mathcal{V}$  is a formula.
2. **false** is a formula.
3. If  $A$  and  $B$  are formulas then  $(A \rightarrow B)$  is a formula.
4. If  $A$  is a formula then  $\circ A$  and  $\square A$  are formulas.

## Basic Language of $\mathcal{L}_{\text{LTL}}(\mathcal{V})$

### Abbreviations:

$\neg, \vee, \wedge, \leftrightarrow, \mathbf{true}$  as in classical logic,

$$\diamond A \equiv \neg \square \neg A.$$

### Note:

$\neg, \circ, \square, \diamond$  are binding stronger than  $\vee, \wedge, \rightarrow, \leftrightarrow$  by definition.

## Kripke Structures

### Definition 10 (Kripke Structure)

Let  $\mathcal{V}$  be a set of atomic propositions.

A **temporal** (or **Kripke**) **structure** for  $\mathcal{V}$  is an infinite sequence

$K = (\eta_0, \eta_1, \eta_2, \dots)$  of mappings

$$\eta_i : \mathcal{V} \rightarrow \{\text{ff}, \text{tt}\}$$

called **states**.

$\eta_0$  is called **initial state** of  $K$ .

see also exercise 7

## Definition of $K_i(F) \in \{\text{ff}, \text{tt}\}$

### Definition 11 (Inductive Definition of $K_i(F) \in \{\text{ff}, \text{tt}\}$ )

1.  $K_i(v) = \eta_i(v)$  for  $v \in \mathcal{V}$ .
2.  $K_i(\mathbf{false}) = \text{ff}$ .
3.  $K_i(A \rightarrow B) = \text{tt} \Leftrightarrow K_i(A) = \text{ff}$  or  $K_i(B) = \text{tt}$ .
4.  $K_i(\bigcirc A) = K_{i+1}(A)$ .
5.  $K_i(\Box A) = \text{tt} \Leftrightarrow K_j(A) = \text{tt}$  for every  $j \geq i$ .

The above definitions induce the following abbreviations:

$$6. K_i(\neg A) = \text{tt} \Leftrightarrow K_i(A) = \text{ff}.$$

$$7. K_i(A \vee B) = \text{tt} \Leftrightarrow K_i(A) = \text{tt} \text{ or } K_i(B) = \text{tt}.$$

$$8. K_i(A \wedge B) = \text{tt} \Leftrightarrow K_i(A) = \text{tt} \text{ and } K_i(B) = \text{tt}.$$

$$9. K_i(A \leftrightarrow B) = \text{tt} \Leftrightarrow K_i(A) = K_i(B).$$

$$10. K(\mathbf{true}) = \text{tt}.$$

$$11. K_i(\diamond A) = \text{tt} \Leftrightarrow K_j(A) = \text{tt} \text{ for some } j \geq i.$$

## Definition 12 (Validity of Formulas)

A formula  $A$  of  $\mathcal{L}_{\text{LTL}}(\mathcal{V})$  is called **valid in the temporal structure**  $K$  for  $\mathcal{V}$  ( $\models_K A$ ) if

$$K_i(A) = \text{tt} \quad \text{for every } i \in \mathbb{N}_0.$$

$A$  is called **valid** ( $\models A$ ) if

$$\models_K A \quad \text{holds for every such } K.$$

see also exercise 8

$A$  is a **consequence of** a set  $\mathcal{F}$  of formulas

( $\mathcal{F} \models A$  or  $B_1, \dots, B_n \models A$  if  $\mathcal{F} = \{B_1, \dots, B_n\}$ ,  $n \geq 1$ ) if

$$\models_K A \quad \text{holds for every } K \text{ with } \models_K B \text{ for all } B \in \mathcal{F}.$$

see also exercise 15

**Lemma 2.1.1**

Let  $K = (\eta_0, \eta_1, \eta_2, \dots)$  and  $K' = (\eta'_0, \eta'_1, \eta'_2, \dots)$  be *temporal structures*,  $i \in \mathbb{N}_0$ .

**a)** If  $K_i(A) = \text{tt}$  and  $K_i(A \rightarrow B) = \text{tt}$  then  $K_i(B) = \text{tt}$ .

**b)** If  $\eta'_j = \eta_{i+j}$  for every  $j \in \mathbb{N}_0$  then  $K'_j(A) = K_{i+j}(A)$  for every  $j \in \mathbb{N}_0$ .

**Theorem 2.1.2**

If  $\models A_1 \wedge \dots \wedge A_n \rightarrow B$  then  $A_1, \dots, A_n \models B$ .

**Theorem 2.1.3**

$A_1, \dots, A_n \models B$  if and only if  $\models \Box A \wedge \dots \wedge \Box A_n \rightarrow B$ .

**Theorem 2.1.4**

If  $\mathcal{F} \models A$  and  $\mathcal{F} \models A \rightarrow B$  then  $\mathcal{F} \models B$ .

## Theorem 2.1.5

If  $\mathcal{F} \models A$  then  $\mathcal{F} \models \bigcirc A$  and  $\mathcal{F} \models \Box A$ . In particular:  $A \models \bigcirc A$  and  $A \models \Box A$ .

see also exercise 9

Some temporal logic formulas and their informal meaning:

$A \rightarrow \bigcirc B$ : “If  $A$  then  $B$  in the next state”,

$A \rightarrow \Box B$ : “If  $A$  then (now and) henceforth  $B$ ”,

$A \rightarrow \Diamond B$ : “If  $A$  then sometime (now or in the future)  $B$ ”,

$\Box(A \rightarrow B)$ : “Whenever (now or) henceforth  $A$  then  $B$  in that state”,

$\Diamond \Box A$ : “Sometime  $A$  will hold permanently, i.e. “ $A$  will be false only finitly often”,

$\Box \Diamond A$ : “For all following states,  $A$  will hold in a later state”, i.e.,

“ $A$  holds infinitely often from now on”.

see also exercise 11

## Temporal Logical Laws

### Definition 13 (Tautological Validity)

A formula of  $\mathcal{L}_{\text{LTL}}$  is called **tautologically valid** if it results from a tautology  $A$  of  $\mathcal{L}_{\text{PL}}$  by consistently replacing the atomic propositions of  $A$  by formulas of  $\mathcal{L}_{\text{LTL}}$ .

### Theorem 2.2.1

*Every tautologically valid formula is valid.*

### Definition 14 (Tautological Consequence)

Let  $A_1, \dots, A_n, B$  ( $n \geq 1$ ) be formulas of  $\mathcal{L}_{\text{LTL}}$ .  $B$  is called a **tautological consequence** of  $A_1, \dots, A_n$  if the formula  $A_1 \rightarrow (A_2 \rightarrow \dots \rightarrow (A_n \rightarrow B) \dots)$  is tautologically valid.

### Theorem 2.2.2

*If  $B$  is a tautological consequence of  $A_1, \dots, A_n$  then  $A_1, \dots, A_n \models B$ .*

## Duality laws

$$(T1) \quad \neg \bigcirc A \leftrightarrow \bigcirc \neg A$$

$$(T2) \quad \neg \Box A \leftrightarrow \Diamond \neg A$$

$$(T3) \quad \neg \Diamond A \leftrightarrow \Box \neg A$$

## Reflexivity laws

$$(T4) \quad \Box A \rightarrow A$$

$$(T5) \quad A \rightarrow \Diamond A$$

## Laws about the “strength” of the operators

$$(T6) \quad \Box A \rightarrow \bigcirc A$$

$$(T7) \quad \bigcirc A \rightarrow \Diamond A$$

$$(T8) \quad \Box A \rightarrow \Diamond A$$

$$(T9) \quad \Diamond \Box A \rightarrow \Box \Diamond A$$

## Idempotency laws

$$(T10) \quad \Box \Box A \leftrightarrow \Box A$$

$$(T11) \quad \Diamond \Diamond A \leftrightarrow \Diamond A$$

### Commutativity laws

$$(T12) \quad \Box \circ A \leftrightarrow \circ \Box A$$

$$(T13) \quad \Diamond \circ A \leftrightarrow \circ \Diamond A$$

### Distributivity laws

$$(T14) \quad \circ(A \rightarrow B) \leftrightarrow \circ A \rightarrow \circ B$$

$$(T15) \quad \circ(A \wedge B) \leftrightarrow \circ A \wedge \circ B$$

$$(T16) \quad \circ(A \vee B) \leftrightarrow \circ A \vee \circ B$$

$$(T17) \quad \Box(A \wedge B) \leftrightarrow \Box A \wedge \Box B$$

$$(T18) \quad \Diamond(A \vee B) \leftrightarrow \Diamond A \vee \Diamond B$$

### Weak distributivity laws

$$(T19) \quad \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$$

$$(T20) \quad \Box A \vee \Box B \rightarrow \Box(A \vee B)$$

$$(T21) \quad (\Diamond A \rightarrow \Diamond B) \rightarrow \Diamond(A \rightarrow B)$$

$$(T22) \quad \Diamond(A \wedge B) \rightarrow \Diamond A \wedge \Diamond B$$

Fixpoint characterizations of  $\Box$  and  $\Diamond$ 

$$(T23) \quad \Box A \leftrightarrow A \wedge \bigcirc \Box A$$

$$(T24) \quad \Diamond A \leftrightarrow A \vee \bigcirc \Diamond A$$

## Monotonicity laws

$$(T25) \quad \Box(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)$$

$$(T26) \quad \Box(A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B)$$

## Frame laws

$$(T27) \quad \Box A \rightarrow (\bigcirc B \rightarrow \bigcirc(A \wedge B))$$

$$(T28) \quad \Box A \rightarrow (\Box B \rightarrow \Box(A \wedge B))$$

$$(T29) \quad \Box A \rightarrow (\Diamond B \rightarrow \Diamond(A \wedge B))$$

## Temporal generalization and particularization laws

$$(T30) \quad \Box(\Box A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$$

$$(T31) \quad \Box(A \rightarrow \Diamond B) \rightarrow (\Diamond A \rightarrow \Diamond B)$$

**Note:**

Laws of the form  $\Box A \rightarrow B$  can also be written as a consequence relationship in the form  $A \vDash B$ , (see theorem 2.1.3)

for example:

$$(T19) \quad A \rightarrow B \vDash \Box A \rightarrow \Box B$$

$$(T30) \quad \Box A \rightarrow B \vDash \Box A \rightarrow \Box B$$

$$(T25) \quad A \rightarrow B \vDash \bigcirc A \rightarrow \bigcirc B$$

$$(T31) \quad A \rightarrow \Diamond B \vDash \Diamond A \rightarrow \Diamond B$$

$$(T27) \quad A \vDash \bigcirc B \rightarrow \bigcirc(A \wedge B)$$

# Axiomatization of LTL

## Axioms

(taut) All tautologically valid formulas,

(ltl1)  $\neg \bigcirc A \leftrightarrow \bigcirc \neg A$ ,

(ltl2)  $\bigcirc(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)$ ,

(ltl3)  $\Box A \rightarrow A \wedge \bigcirc \Box A$ .

## Rules

(mp)  $A, A \rightarrow B \vdash B$ ,

(nex)  $A \vdash \bigcirc A$ ,

(ind)  $A \rightarrow B, A \rightarrow \bigcirc A \vdash A \rightarrow \Box B$ .

## Theorem 2.3.1

### (Soundness Theorem for $\Sigma_{\text{LTL}}$ )

Let  $A$  be a formula and  $\mathcal{F}$  a set of formulas. If  $\mathcal{F} \vdash A$  then  $\mathcal{F} \models A$ .

In particular: If  $\vdash A$  then  $\models A$ .

## Theorem 2.3.2

If  $B$  is a tautological consequence of  $A_1, \dots, A_n$  then  $A_1, \dots, A_n \vdash B$ .

We use this theorem as a **new rule**:

(prop)  $A_1, \dots, A_n \vdash B$  if  $B$  is a tautological consequence of  $A_1, \dots, A_n$ .

## Example 6 (Chaining Rule)

$A \rightarrow B, B \rightarrow C \vdash A \rightarrow C$

see also exercises 10 and 12

## Example 7 (Derivation of the “opposite direction” of (ltl2))

$$(ltl2') \quad (\bigcirc A \rightarrow \bigcirc B) \rightarrow \bigcirc(A \rightarrow B)$$

- |     |  |                       |
|-----|--|-----------------------|
| (1) | $\neg(A \rightarrow B) \rightarrow A$  | <i>(taut)</i>         |
| (2) | $\bigcirc(\neg(A \rightarrow B) \rightarrow A)$  | <i>(nex),(1)</i>      |
| (3) | $\bigcirc(\neg(A \rightarrow B) \rightarrow A) \rightarrow (\bigcirc\neg(A \rightarrow B) \rightarrow \bigcirc A)$ | <i>(ltl2)</i>         |
| (4) | $\bigcirc\neg(A \rightarrow B) \rightarrow \bigcirc A$   | <i>(mp),(2),(3)</i>   |
| (5) | $\neg\bigcirc(A \rightarrow B) \leftrightarrow \bigcirc\neg(A \rightarrow B)$                                      | <i>(ltl1)</i>         |
| (6) | $\neg\bigcirc(A \rightarrow B) \rightarrow \bigcirc\neg(A \rightarrow B)$  | <i>(prop),(5)</i>     |
| (7) | $\neg\bigcirc(A \rightarrow B) \rightarrow \bigcirc A$   | <i>(prop),(6),(4)</i> |

*continued on the next page*

- (8)  $\neg(A \rightarrow B) \rightarrow \neg B$  *(taut)*
- (9)  $\neg\bigcirc(A \rightarrow B) \rightarrow \bigcirc\neg B$  *from (8) in the same way as (7) from (1)*
- (10)  $\bigcirc\neg B \rightarrow \neg\bigcirc B$  *(prop),(ltl1)*
- (11)  $\neg\bigcirc(A \rightarrow B) \rightarrow \neg\bigcirc B$  *(prop),(9),(10)*
- (12)  $\neg\bigcirc(A \rightarrow B) \rightarrow \neg(\bigcirc A \rightarrow \bigcirc B)$  *(prop),(7),(11)*
- (13)  $(\bigcirc A \rightarrow \bigcirc B) \rightarrow \bigcirc(A \rightarrow B)$  *(prop),(12)*

**Example 8 (Derivation of the “opposite direction” of (ltl3))**

(ltl3')  $A \wedge \bigcirc \Box A \rightarrow A$

- |     |   |   |
|-----|---|---|
| (1) | $A \wedge \bigcirc \Box A \rightarrow A$  | ( <i>taut</i> )                                     |
| (2) | $\Box A \rightarrow A \wedge \bigcirc \Box A$   | ( <i>ltl3</i> )                                     |
| (3) | $\bigcirc \Box A \rightarrow \bigcirc(A \wedge \bigcirc \Box A)$                        | ( <i>nex</i> ), ( <i>ltl2</i> ), ( <i>mp</i> ), (2) |
| (4) | $A \wedge \bigcirc \Box A \rightarrow \bigcirc(A \wedge \bigcirc \Box A \rightarrow A)$ | ( <i>prop</i> ), (3)                                |
| (5) | $A \wedge \bigcirc \Box A \rightarrow A$  | ( <i>ind</i> ), (1), (4)                            |

## Some Useful Derived Rules

The following two rules are useful variants of the induction rule (ind):

$$(ind1) \quad A \rightarrow \bigcirc A \vdash A \rightarrow \Box A$$

$$(ind2) \quad A \rightarrow B, B \rightarrow \bigcirc B \vdash A \rightarrow \Box B$$

Next we show two rules the first of which is the analogy of (nex) for  $\Box$ :

$$(alw) \quad A \vdash \Box A$$

$$(som) \quad A \rightarrow \bigcirc B \vdash A \rightarrow \Diamond B$$

## Example 9 (Derivation of (T15'))

$$(T15') \quad \bigcirc A \wedge \bigcirc B \rightarrow \bigcirc(A \wedge B)$$

- |     |  |                              |
|-----|--|------------------------------|
| (1) | $\bigcirc(A \rightarrow \neg B) \rightarrow (\bigcirc A \rightarrow \bigcirc \neg B)$          | <i>(ltl2)</i>                |
| (2) | $\bigcirc(A \rightarrow \neg B) \rightarrow (\bigcirc A \rightarrow \neg \bigcirc B)$          | <i>(prop), (ltl1), (1)</i>   |
| (3) | $\neg(\bigcirc A \rightarrow \neg \bigcirc B) \rightarrow \neg \bigcirc(A \rightarrow \neg B)$ | <i>(prop), (2)</i>           |
| (4) | $\neg(\bigcirc A \rightarrow \neg \bigcirc B) \rightarrow \bigcirc \neg(A \rightarrow \neg B)$ | <i>(prop), (ltl1), (3)</i>   |
| (5) | $\bigcirc A \wedge \bigcirc B \rightarrow \bigcirc(A \wedge B)$                                | <i>(4) in other notation</i> |

## Theorem 2.3.3

### (Deduction theorem of LTL)

Let  $A, B$  be formulas,  $\mathcal{F}$  a set of formulas. If  $\mathcal{F} \cup \{A\} \vdash B$  then  $\mathcal{F} \vdash \Box A \rightarrow B$ .

### Note:

The Deduction Theorem of classical propositional logic:

$$\text{If } \mathcal{F} \cup \{A\} \vdash B \text{ then } \mathcal{F} \vdash A \rightarrow B$$

does not hold generally in LTL.

### Special cases of the Deduction Theorem:

- 1) If  $A \vdash B$  then  $\vdash \Box A \rightarrow B$ .
- 2) If  $A_1, \dots, A_n \vdash B$  then  $\vdash \Box A_1 \wedge \dots \wedge \Box A_n \rightarrow B$ .

## Theorem 2.3.4

Let  $A, B$  be formulas,  $\mathcal{F}$  a set of formulas. If  $\mathcal{F} \vdash \Box A \rightarrow B$  then  $\mathcal{F} \cup \{A\} \vdash B$ .

## Remarks

1. LTL is not completely axiomatizable.
2.  $\Sigma_{\text{LTL}}$  is **weakly complete** in the following sense:

If  $A$  is a formula and  $\mathcal{F}$  a finite set of formulas then:

$$\mathcal{F} \models A \Rightarrow \mathcal{F} \vdash A$$

In particular:

$$\models A \Rightarrow \vdash A$$

see also exercise 13

3. As a consequence, we may use all laws (T1), (T2), ... as derivable formulas and rules, respectively.

---

# Chapter 3

## Extensions of LTL

## Binary Temporal Operators

$A$  **until**  $B$ : “There is a (strictly) subsequent state in which  $B$  holds, and  $A$  holds until that state”,

$A$  **unless**  $B$ : “If there is a (strictly) subsequent state in which  $B$  holds then  $A$  holds until that state or else  $A$  holds permanently”,

$A$  **atnext**  $B$ : “ $A$  holds in the first (strictly) subsequent state in which  $B$  holds (if this state exists)”.

$A$  **before**  $B$ : “If there is a (strictly) subsequent state in which  $B$  holds then  $A$  holds before that state.”

## Formalization

Given a temporal structure  $K$  and  $i \in \mathbb{N}_0$ :

see also exercise 14 and 16

- $K_i(A \text{ **until** } B) = \text{tt} \iff K_j(B) = \text{tt}$  for some  $j > i$  and  
 $K_k(A) = \text{tt}$  for every  $k, i < k < j$ .
- $K_i(A \text{ **unless** } B) = \text{tt} \iff K_j(B) = \text{tt}$  for some  $j > i$  and  
 $K_k(A) = \text{tt}$  for every  $k, i < k < j$   
 or  
 $K_k(A) = \text{tt}$  for every  $k > i$ .
- $K_i(A \text{ **atnext** } B) = \text{tt} \iff K_j(B) = \text{ff}$  for every  $j > i$  or  
 $K_k(A) = \text{tt}$  for the smallest  $k > i$  such that  $K_k(B) = \text{tt}$ ,
- $K_i(A \text{ **before** } B) = \text{tt} \iff$  for every  $j > i$  with  $K_j(B) = \text{tt}$   
 there is some  $k, i < k < j$ , such that  $K_k(A) = \text{tt}$ .

## Relationships between the Operators

### 1. Strong and weak operators:

$$(T33) \quad A \text{ until } B \leftrightarrow \bigcirc \diamond B \wedge A \text{ unless } B$$

$$(T34) \quad A \text{ unless } B \leftrightarrow A \text{ until } B \vee \bigcirc \square A$$

### 2. Mutual expressibility (of weak operators)

$$(T35) \quad A \text{ unless } B \leftrightarrow B \text{ atnext } (A \rightarrow B)$$

$$(T36) \quad A \text{ before } B \leftrightarrow \neg B \text{ atnext } (A \vee B)$$

$$(T37) \quad A \text{ atnext } B \leftrightarrow \neg B \text{ unless } (A \wedge B)$$

$$(T38) \quad A \text{ atnext } B \leftrightarrow B \text{ before } (\neg A \wedge B)$$

### 3. Expressibility of $\circ$ and $\square$ :

$$(T39) \quad \circ A \leftrightarrow A \text{ atnext true}$$

$$(T40) \quad \square A \leftrightarrow A \wedge \text{false atnext } \neg A$$

### 4. Fixpoint characterizations:

$$(T41) \quad A \text{ until } B \leftrightarrow \circ B \vee \circ(A \wedge A \text{ until } B)$$

$$(T42) \quad A \text{ unless } B \leftrightarrow \circ B \vee \circ(A \wedge A \text{ unless } B)$$

$$(T43) \quad A \text{ atnext } B \leftrightarrow \circ(B \rightarrow A) \wedge \circ(\neg B \rightarrow A \text{ atnext } B)$$

$$(T44) \quad A \text{ before } B \leftrightarrow \circ\neg B \wedge \circ(A \vee A \text{ before } B)$$

5. More laws for **atnext**:

$$(T45) \quad \Box A \rightarrow A \mathbf{atnext} B$$

$$(T46) \quad \bigcirc(A \mathbf{atnext} B) \leftrightarrow \bigcirc A \mathbf{atnext} \bigcirc B$$

$$(T47) \quad (A \wedge B) \mathbf{atnext} C \leftrightarrow A \mathbf{atnext} C \wedge B \mathbf{atnext} C$$

$$(T48) \quad (A \vee B) \mathbf{atnext} C \leftrightarrow A \mathbf{atnext} C \vee B \mathbf{atnext} C$$

$$(T49) \quad A \mathbf{atnext} (B \vee C) \rightarrow A \mathbf{atnext} B \vee A \mathbf{atnext} C$$

$$(T50) \quad \Box(A \rightarrow B) \rightarrow (A \mathbf{atnext} C \rightarrow B \mathbf{atnext} C)$$

$$(T51) \quad \Box A \rightarrow (B \mathbf{atnext} C \rightarrow (A \wedge B) \mathbf{atnext} (A \wedge C))$$

## Language Extension of $\mathcal{L}_{LTL}$

Extension of  $\mathcal{L}_{LTL}$  to  $\mathcal{L}_{LTL}^b$  ( $\mathcal{L}_{LTL}$  with additional binary operators)

**op**  $\in$  {**until**, **unless**, **atnext**, **before**, ...}

- Extend **alphabet** of  $\mathcal{L}_{LTL}$  by **op**.
- Extend **definition of formulas**:
  5. If  $A$  and  $B$  are formulas then  $A$  **op**  $B$  is a formula.
- Extend **definition of semantics** by  $K_i(A \text{ op } B) = \dots$  (see above).

see also exercise 18

## Axiomatization

A formal system for **LTL + b** (Linear Temporal Logic with binary operators) is obtained by extending  $\Sigma_{\text{LTL}}$  by axioms for the **selected binary operator op** ( $\rightsquigarrow$  formal system  $\Sigma_{\text{LTL}}^b$ ).

In any case, **two axioms** for the respective operator are necessary:

- “**strong**”/“**weak**” characterization
- **fixpoint characterization** (T41)/.../(T44)

**atnext** as basic operator:

$$\text{(atn1)} \quad \bigcirc \square \neg B \rightarrow A \text{ atnext } B$$

$$\text{(atn2)} \quad A \text{ atnext } B \leftrightarrow \bigcirc(B \rightarrow A) \wedge \bigcirc(\neg B \rightarrow A \text{ atnext } B)$$

**until** as basic operator:

$$(unt1) \quad A \text{ **until** } B \rightarrow \bigcirc \diamond B$$

$$(unt2) \quad A \text{ **until** } B \leftrightarrow \bigcirc B \vee \bigcirc(A \wedge A \text{ **until** } B)$$

**unless** as basic operator:

see also exercise 17

$$(unl1) \quad \bigcirc \square A \rightarrow A \text{ **unless** } B$$

$$(unl2) \quad A \text{ **unless** } B \leftrightarrow \bigcirc B \vee \bigcirc(A \wedge A \text{ **unless** } B)$$

**before** as basic operator:

$$(bef1) \quad \bigcirc \square \neg B \rightarrow A \text{ **before** } B$$

$$(bef2) \quad A \text{ **before** } B \leftrightarrow \bigcirc \neg B \wedge \bigcirc(A \vee A \text{ **before** } B))$$

## Induction Rules for weak binary operators

(indatnext)  $A \rightarrow \circ(C \rightarrow B) \wedge \circ(\neg C \rightarrow A) \vdash A \rightarrow B$  **atnext**  $C$

(indunless)  $A \rightarrow \circ C \vee \circ(A \wedge B) \vdash A \rightarrow B$  **unless**  $C$

(indbefore)  $A \rightarrow \circ\neg C \wedge \circ(A \vee B) \vdash A \rightarrow B$  **before**  $C$

## Past Operators

Extension of  $\mathcal{L}_{\text{LTL}}$  to  $\mathcal{L}_{\text{LTL}}^{\text{P}}$  ( $\mathcal{L}_{\text{LTL}}$  with additional (unary) “past” operators):

$\ominus$ : weak previous operator

$\boxminus$ : has-always-been operator

Informal meaning:  $\ominus A$ : “ $A$  held in the previous state”,

$\boxminus A$ : “ $A$  held in all past states (including the present one)”.

Extended **definition of formulas**: If  $A$  is a formula then  $\ominus A$  and  $\boxminus A$  are formulas.

Semantics ( $K$  temporal structure,  $i \in \mathbb{N}_0$ ):

- $K_i(\ominus A) = \text{tt} \Leftrightarrow$  if  $i > 0$  then  $K_{i-1}(A) = \text{tt}$ .
- $K_i(\boxminus A) = \text{tt} \Leftrightarrow K_j(A) = \text{tt}$  for every  $j \leq i$ .

## More Operators (Abbreviations)

$$\ominus A \equiv \neg \ominus \neg A \quad (\text{strong previous operator})$$

$$\diamond A \equiv \neg \boxminus \neg A \quad (\text{once operator})$$

Semantics:

$$K_i(\ominus A) = \text{tt} \Leftrightarrow i > 0 \text{ and } K_{i-1}(A) = \text{tt},$$

$$K_i(\diamond A) = \text{tt} \Leftrightarrow K_j(A) = \text{tt} \text{ for some } j \leq i.$$

## Valid Formulas

$$(P1) \quad \ominus A \rightarrow \neg \ominus \text{false}$$

$$(P5) \quad \ominus (A \rightarrow B) \leftrightarrow \ominus A \rightarrow \ominus B$$

$$(P2) \quad \ominus \neg A \rightarrow \neg \ominus A$$

$$(P6) \quad \ominus (A \wedge B) \leftrightarrow \ominus A \wedge \ominus B$$

$$(P3) \quad A \rightarrow \ominus \circ A$$

$$(P7) \quad \ominus (A \wedge B) \leftrightarrow \ominus A \wedge \ominus B$$

$$(P4) \quad A \rightarrow \circ \ominus A$$

## Axiomatization

Extension of  $\Sigma_{\text{LTL}}$  to  $\Sigma_{\text{LTL}}^{\text{P}}$  by additional axioms and rules:

$$\text{(pltl1)} \quad \ominus \neg A \rightarrow \neg \ominus A$$

$$\text{(pltl4)} \quad \diamond \ominus \mathbf{false}$$

$$\text{(pltl2)} \quad \ominus (A \rightarrow B) \rightarrow (\ominus A \rightarrow \ominus B)$$

$$\text{(pltl5)} \quad A \rightarrow \ominus \circ A$$

$$\text{(pltl3)} \quad \boxplus A \rightarrow A \wedge \ominus \boxplus A$$

$$\text{(pltl6)} \quad A \rightarrow \circ \ominus A$$

$$\text{(prev)} \quad A \vdash \ominus A$$

$$\text{(indpast)} \quad A \rightarrow B, A \rightarrow \ominus A \vdash A \rightarrow \boxplus B$$

$\Sigma_{\text{LTL}}^{\text{P}}$  is **sound** and **complete**.

## Example 10 (Derivation of (P1))

- |     |   |  |
|-----|---|--|
| (1) | $\mathbf{false} \rightarrow \neg A$                 | $(\mathit{taut})$                        |
| (2) | $\ominus (\mathbf{false} \rightarrow \neg A)$       | $(\mathit{prev}), (1)$                   |
| (3) | $\ominus \mathbf{false} \rightarrow \ominus \neg A$ | $(\mathit{pltl2}), (\mathit{prop}), (2)$ |
| (4) | $\ominus A \rightarrow \neg \ominus \mathbf{false}$ | $(\mathit{prop}), (3)$                   |