Information Systems Analysis

# Temporal Logic and Timed Automata

(1)

## Introduction to temporal logic



*version 2.2*

# Program of lectures

1. Introduction to temporal logic
2. LTL logic
3. CTL logic
4. System model verification
5. UPPAAL timed automata
6. System model verification in UPPAAL
7. NuSMV timed automata
8. System model verification in NuSMV
9. Other kinds of temporal logic
10. Temporal logic and temporal databases

# Contents of the lecture

## Logic

- What is logic?
- Classical sentential logic
- Modal logic

# Contents of the lecture

## Time

- What is time?
- Dense and discrete time
- Finite and infinite time
- Linear, branching and parallel time
- Point and interval time

# Contents of the lecture

## Temporal logic

- What is temporal logic?
- Selected kinds of temporal logic
- Application of temporal logic

# Logic

- What is logic?
- Classical sentential logic
- Modal logic

# Logic

## What is logic?

- science of argumentation *(from Greek λογική /logiké/)*
- one of basic branches of the maths
  *(remaining are: the set theory and the arithmetic)*
- a mathematical description of reality
- a tool to distinguish between truth and falsehood
  *(as defined by Awerroes in XII century)*
- science of reasoning, and the art of reasoning
  *(as defined by Richard Whately in XIX century)*
- science of the most general laws of truth
  *(as defined by Gottlob Frege in XIX/XX century)*

# Logic

## Classical sentential logic

- is a set of logic laws;
- gives a logic state (*truth* / *falsity*) for every sentence,
  e.g. the sentence “October is a month.” is true;
- allows to reason about veracity* of a sentence,
  e.g. “If x and y are numbers, then their sum is a number too”.

* A sentence is true, if it describes an actual state of things.

# Logic

## Modal logic

- is the classical sentential logic mathematically specified.

Let AP be a set of atomic propositions

*An **Atomic proposition** is indivisible into other propositions.*
*E.g. "Water is wet." is not divisible into other sentences.*

A formula is generated by rules S1 and S2:

**S1:** every proposition $p \in$ AP is a formula;

**S2:** if $p$ and $q$ are formulas, then $p \wedge q$ and $\neg p$ are formulas too

(other classic logic connectives are also accessible).

# Modal logic

Basic and derivative logic relations:

| relation | description | meaning |
|---|---|---|
| conjunction | $p \wedge q$ | $p$ is true<br>and $q$ is true |
| negation | $\neg p$ | $p$ is not true |
| disjunction | $p \vee q$ | $\neg (\neg p \wedge \neg q)$ |
| implication | $p \Rightarrow q$ | $\neg p \vee q$ |
| equality | $p \Leftrightarrow q$ | $(p \Rightarrow q) \wedge (q \Rightarrow p)$ |

$p$ and $q$ are formulas

# Time

- What is time?
- Dense and discrete time
- Finite and infinite time
- Linear, branching and parallel time
- Point and interval time

# Time

## What is time?

- nonspatial continuum, in which events occur in an irreversible sequence from the past through the present to the future;
- one of the dimensions of the space–time continuum in physics.

Let $t_0$, $t_1$, $t_2 \in$ *time* be moments* of time.

** A **moment** is a zero-length time, represented e.g. by a point on a time axis.*

# Time

## Dense and discrete time

**Dense**: $(\forall t_0,t_1)\ (\exists t_2)\ (t_0 < t_2 < t_1)$

— for every two moments there exists a moment between them.

**Discrete**: $(\exists t_0,t_1)\ (\neg\exists t_2)\ (t_0 < t_2 < t_1)$

— there exist two such moments, that there does not exist any moment between them.

# Time

## Finite and infinite time

**Left–sided finite**: $(\exists t_0)\ (\neg\exists t_1)\ (t_1 < t_0)$

— the past is limited to some moment (e.g. the present one);
the future is infinite.

**Right–sided finite**: $(\exists t_0)\ (\neg\exists t_1)\ (t_1 > t_0)$

— the past is infinite;
the future is limited to some moment (e.g. the present one);

**Both–sided finite**: $(\exists t_0,t_1)\ (\neg\exists t_2)\ (t_2 < t_0 \vee t_2 > t_1)$

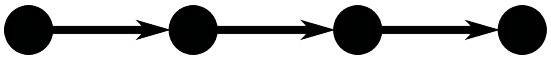
— the past and the future are limited to some moments;

**Infinite**: $(\forall t_0)\ (\exists t_1,t_2)\ (t_1 < t_0 < t_2)$

— the past and the future are unlimited.

# Time

## Linear, branching and parallel time

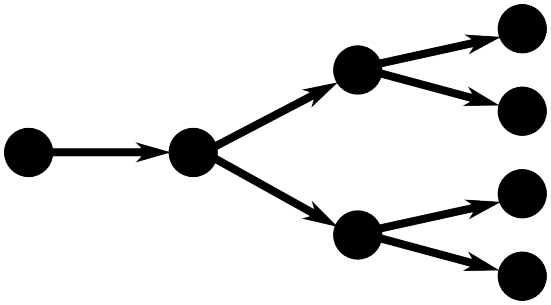


**Linear**: $(\forall t_0,t_1)\ (t_0 < t_1 \ \vee\ t_0 = t_1 \ \vee\ t_0 > t_1)$

— there exists only one variant of the flow of time.

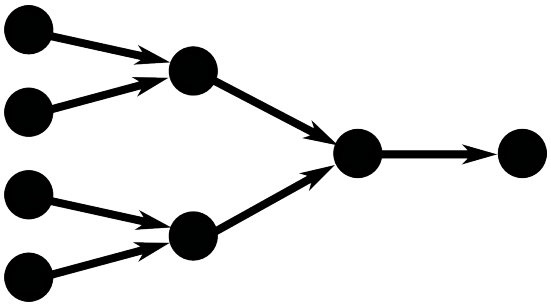
**Branching**:

— there exist different variants of the flow of time, having a common part with each other:



- left–sided linearity:

$(\forall t_0,t_1,t_2)\ (t_1< t_0 \wedge t_2< t_0 \Rightarrow t_1< t_2 \vee t_1= t_2 \vee t_1> t_2)$



- right–sided linearity:

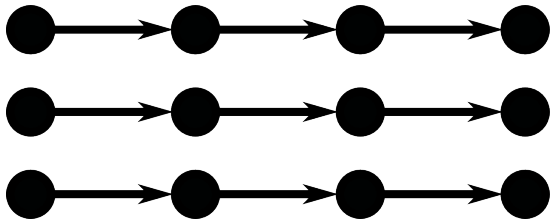
$(\forall t_0,t_1,t_2)\ (t_0< t_1 \wedge t_0< t_2 \Rightarrow t_1< t_2 \vee t_1= t_2 \vee t_1> t_2)$

# Linear, branching and parallel time

**Parallel**:

— there exist different variants of the flow of time, having no common part with each other;

— time is both left–sided and right–sided linear.



# Time

## Point and interval time

**Point**:

— the time structure consists of points only.

**Interval**:

— the time structure consists of intervals only.

# Temporal logic

- What is temporal logic?
- Selected kinds of temporal logic
- Application of temporal logic

# Temporal logic

## What is temporal logic?

- allows reasoning with time considered

  *(reasoning over time domain),*
- assigns the value *true* / *false* to modal logic propositions,
- places the value *true* / *false* in a time structure.

# Temporal logic

## Selected kinds of temporal logic:

- **LTL** (*Linear Temporal Logic*)

  — time: discrete, left-sided finite, linear, point-like;

- **CTL** (*Computation Tree Logic*)

  — time: discrete, left-sided finite, branching (left-sided linear), point-like;

- **RTCTL** (*Real-Time Computation Tree Logic*)

  — a version of CTL, where values of time are given quantitatively as constants;

- **PRTCTL** (*Parametrised Real-Time Computation Tree Logic*)

  — a version of CTL, where values of time are given quantitatively as variables.

# Temporal logic

## Selected kinds of temporal logic:

- **TCTL** (*Timed Computation Tree Logic*)

  — a version of CTL, where values of time are given quantitatively as constants;

- **CTL*** (*Full Branching Computation Tree Logic*)

  — a version of CTL of a greater expression, including expression of LTL;

- **ITL** (*Interval Temporal Logic*)

  — time: discrete, finite or infinite, linear, interval-like;

- **DC** (*Duration Calculus*)

  — a version of ITL, where truth of an expression is measured in a time interval.

# Application of temporal logic

- Mainly in modelling and verifying of systems that are:
  - concurrent

    (they consist of processes running in parallel, having access to common data, potentially interacting with each other);
  - reactive

    (they work in the action-reaction manner, in relation to arising circumstances, e.g.: controlling an electric reactor, an engine in a machine or the air traffic).

# The end

**Literature:**


- E.A. Emerson “Temporal and modal logic”, 1995
- R. Klimek “*Zastosowanie logiki temporalnej w specyfikacji i weryfikacji oprogramowania – w stronę czasu rzeczywistego”*, in: “Wykłady zaproszone. Systemy Czasu Rzeczywistego 2000”, 2001