UNIVERSITÀ
DI TRENTO
Department of
Information Engineering and Computer Science

Automated Reasoning and **Formal Verification**

Laboratory 11

Gabriele Masina
gabriele.masina@unitn.it
https://github.com/masinag/arfv2025

Università di Trento

May 20, 2025

These slides are derived from those by Stefano Tonetta, Alberto Griggio, Silvia Tomasi, Thi Thieu Hoa Le, Alessandra Giordani, Patrick Trentin, Giuseppe Spallitta for FM lab 2005-2024.

Timed Systems

Real time systems

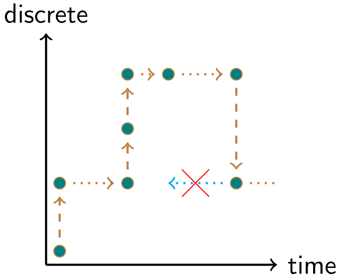
- Correctness depends both on the logical result and the time required to compute it
- Safety-critical domains: defense, transportation, health-care, space, avionics, etc.

Timed Transition System (TTS)

- Transitions are either discrete or time-elapses
- All clocks increase equally in time-elapses
- Model Checking for TTS is undecidable

Timed Automata (TA)

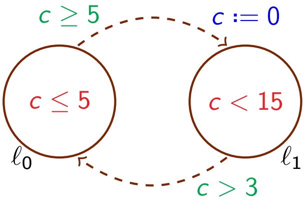
- Decidable restriction of TTS
- Finite Time Abstraction:
  clocks compared only to constants



Timed Systems: Representation

Timed Automata (TA)

- Explicit graph representation:
  discrete states (nodes) and transitions (edges).
- Symbolic representation of temporal aspects via (convex) constraints:
  location invariants, transition guards and resets.



Symbolic TTS

- Formulas represent sets of states: $p := \{s \mid s \models p\}$
- Symbolic transitions $\varphi(X, X')$
- There is a discrete transition $s_0 \rightarrow s_1$ iff
  $s_0(X), s_1(X') \models \varphi(X, X')$

$\Longrightarrow$ can be described in (timed) nuXmv!

$$
\begin{aligned}
&(l = \ell_0) \rightarrow (c \leq 5) && \wedge \\
&(l = \ell_1) \rightarrow (c < 15) && \wedge \\
&(l = \ell_1 \wedge l' = \ell_0) \rightarrow (c > 3) && \wedge \\
&(l = \ell_0 \wedge l' = \ell_1) \rightarrow \\
&\qquad ((c \geq 5) \wedge (c' = 0))
\end{aligned}
$$

Outline

1. Timed nuXmv
2. Timed and Infinite Traces
3. Exercises
4. Homework

nuXmv for Timed System: Architecture[1]

Timed nuXmv $M_t \models \varphi_t$

Timed traces

reduction to timed
$(M_t \models \varphi_t) \iff (M \models \varphi)$

trace timing
$(\sigma_t \in M_t) \iff (\sigma_t \in M_t)$

nuXmv $M \models \varphi$

Traces

[1] Alessandro Cimatti et al. "Extending nuXmv with Timed Transition Systems and Timed Temporal Properties". In: *CAV 2019*

Timed nuXmv: Input Language [1/5]

Overview

Extension of nuXmv for real-time system modelling and verification:

- Allows for modelling Timed Transition Systems with continuous time domain
- Enables specification and verification of timed properties

Timed nuXmv: Input Language [2/5]

Time Domain and Clock Variables

- Must start with `@TIME_DOMAIN` continuous;
- New variable type: `clock`
- `clock` variables increase uniformly with time
- Built-in `clock` variable: `time`

System Specification

`INIT`	: Specifies initial conditions
`TRANS`	: Constrains discrete behavior only
`INVAR`	: Defines invariant conditions. Clocks allowed in invariants with shape: `(no clock expr) -> (convex clock expr);` (convex expression: conjunction of atoms)

Timed nuXmv: Input Language [4/5]

Timed Transitions

- System evolution alternates between:
  - **discrete** transitions: instantaneous state changes
  - **timed** transitions: passage of time where all clocks increase by the same amount
- `URGENT` conditions (time-freeze): when one of the `URGENT` conditions is satisfied, only discrete transitions are allowed
- `noncontinuous` type modifier: allows variables to change their value during timed transitions

Timed nuXmv: Input Language [5/5]

Specifying Properties

Specifications in **Metric Temporal Logic**[a] ($\text{MTL}_{0,\infty}$):

- LTL operators: `X`, `Y` (yesterday) `U`, `S` (since), `F`, `G`
- Bounded LTL operators: `F[a,b] p`, `G[a,b] p` (eventually/always if *time* $\in [a, b]$)
- Different operators to refer to the discrete next (`X`) and timed next (`X~`), and symmetrically for the past (`Y` and `Y~`)
- Operators to get the value of expr the next/last time an p will hold/held:

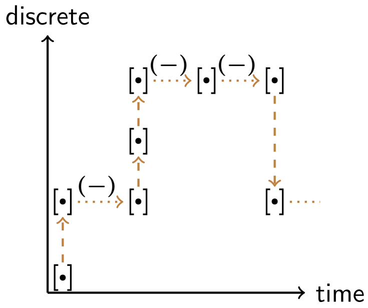
  `time_until(p)` is the time until p will hold
  `time_since(p)` is the time since p held
  `expr @F~ p` is the value expr will have the next time p will hold.
  `expr @O~ p` is the value expr had the last time p held

[a] https://en.wikipedia.org/wiki/Metric_temporal_logic

Timed nuXmv: Untiming

Timed to Untimed Model

- `clock` symbols and `time`: variables of type `real`
- δ: the amount of time elapse for every transition (continuous positive variable)
- ι: prescribes the alternation of singular $[\bullet]$ and open $(-)$ time intervals



Timed nuXmv: Untiming

Properties Rewriting

MTL *fragment* $\boxed{F_{[0,5]}\ p}$

↓ rewrite

LTL *timed*

$$\boxed{\begin{array}{l} ((\neg p U p) \wedge time\_until(p) \leq 5) \vee \\ ((\neg p U \tilde{X} p) \wedge time\_until(p) < 5) \end{array}}$$

↓ untime

LTL *untimed*

$$\boxed{\begin{array}{l} ((\neg p U p) \wedge (time@\tilde{F} p) - time \leq 5) \vee \\ ((\neg p U((\neg \iota \wedge p) \vee X(\neg \iota \wedge p))) \wedge (time@\tilde{F} p) - time < 5) \end{array}}$$

Outline

From Untimed Model Execution to Timed Trace

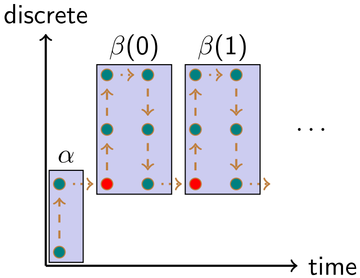
Issue

- nuXmv traces must have lasso shape: $\alpha\beta^\omega$, with α and β sequences of states.
  $\implies$ complete for finite state systems.
- In timed and infinite state systems they may not exists!

Solution

Lasso-shaped traces with diverging variables $\alpha\beta(i)^\omega$:

- Some variables repeat (like in lasso-shaped traces)
- Other variables diverge as a function of previous state: $s_i(v) = f_v(s_{i-1})$
  e.g. $next(time) := time + \delta$



How to Run: Model [1/3]

`./nuXmv -time -int` start nuXmv interactively and enable commands for timed models.

`go_time` process the model.

`write_untimed_model` dump SMV model corresponding to the input timed system.

How to Run: Verify [2/3]

`timed_check_invar` check invariants

`timed_check_ltlspec` check LTL

Mostly the same command line options of the corresponding commands for untimed models.

How to Run: Simulation and Traces [3/3]

`timed_pick_state` pick initial state.

`timed_simulate` simulate the model starting from a given state.

Semantics of Temporal Operators

Formally nuXmv uses a super-dense weakly-monotonic time model $T \subset \mathbb{N} \times \mathbb{R}_0^+$.

Time points are pairs $\langle i, r \rangle$ with $i \in \mathbb{N}$ (step count) and $r \in \mathbb{R}_0^+$ (time).

We say that $\langle i, r \rangle < \langle i', r' \rangle$ iff $i < i'$ or $i = i'$ and $r < r'$.

Semantics of Temporal Operators

$\sigma, t \models \phi$ is defined recursively on the structure of ϕ:

$\sigma, t \models \phi_1 U \phi_2$	iff there exists $t' \geq t$, $\sigma, t' \models \phi_2$ and for all $t'', t \leq t'' < t'$, $\sigma, t'' \models \phi_1$
$\sigma, t \models \phi_1 S \phi_2$	iff there exists $t' \leq t$, $\sigma, t' \models \phi_2$ and for all $t'', t' < t'' \leq t$, $\sigma, t'' \models \phi_1$
$\sigma, t \models X\phi$	iff there exists $t' > t$, $\sigma, t' \models \phi$ and there exists no $t'', t < t'' < t'$
$\sigma, t \models \tilde{X}\phi$	iff for all $t' > t$, there exists $t'', t < t'' < t'$, $\sigma, t'' \models \phi$
$\sigma, t \models Y\phi$	iff $t > 0$ and there exists $t' < t$, $\sigma, t' \models \phi$ and there exists no $t'', t' < t'' < t$
$\sigma, t \models \tilde{Y}\phi$	iff $t > 0$ and for all $t' < t$, there exists $t'', t' < t'' < t$, $\sigma, t'' \models \phi$

Usual definitions for predicates, conjunction and negation apply.

LTL- MTL Properties [1/2]

Let b be a boolean symbol. Are the following properties true or false?

$\tilde{Y}\top$:

$(\neg Xb) \rightarrow (X\neg b)$:

$(\neg\tilde{X}\ b) \rightarrow (\tilde{X}\neg b)$:

$(X\neg b) \rightarrow (\neg Xb)$:

$(\tilde{X}\neg b) \rightarrow (\neg\tilde{X}b)$:

$(G\tilde{X}\top) \rightarrow ((Gb) \vee (G\neg b))$:

LTL- MTL Properties [1/2]

Let b be a boolean symbol. Are the following properties true or false?

$\tilde{Y}\top$: false in the initial state.

$(\neg X b) \rightarrow (X \neg b)$:

$(\neg \tilde{X}\ b) \rightarrow (\tilde{X} \neg b)$:

$(X \neg b) \rightarrow (\neg X b)$:

$(\tilde{X} \neg b) \rightarrow (\neg \tilde{X} b)$:

$(G \tilde{X} \top) \rightarrow ((G b) \vee (G \neg b))$:

LTL- MTL Properties [1/2]

Let b be a boolean symbol. Are the following properties true or false?

$\tilde{Y}\top$: false in the initial state.

$(\neg Xb) \rightarrow (X\neg b)$: false, the first one holds in every time elapse, the second one holds only in discrete steps where $\neg b$ holds in the next state.

$(\neg \tilde{X}\ b) \rightarrow (\tilde{X}\neg b)$:

$(X\neg b) \rightarrow (\neg Xb)$:

$(\tilde{X}\neg b) \rightarrow (\neg \tilde{X} b)$:

$(G\tilde{X}\top) \rightarrow ((Gb) \vee (G\neg b))$:

LTL- MTL Properties [1/2]

Let b be a boolean symbol. Are the following properties true or false?

$\tilde{Y}\top$: false in the initial state.

$(\neg Xb) \rightarrow (X\neg b)$: false, the first one holds in every time elapse, the second one holds only in discrete steps where $\neg b$ holds in the next state.

$(\neg \tilde{X}\ b) \rightarrow (\tilde{X}\neg b)$: false, as above but for time elapses.

$(X\neg b) \rightarrow (\neg Xb)$:

$(\tilde{X}\neg b) \rightarrow (\neg \tilde{X} b)$:

$(G\tilde{X}\top) \rightarrow ((Gb) \vee (G\neg b))$:

LTL- MTL Properties [1/2]

Let b be a boolean symbol. Are the following properties true or false?

$\tilde{Y}\top$: false in the initial state.

$(\neg X b) \rightarrow (X \neg b)$: false, the first one holds in every time elapse, the second one holds only in discrete steps where $\neg b$ holds in the next state.

$(\neg \tilde{X}\ b) \rightarrow (\tilde{X} \neg b)$: false, as above but for time elapses.

$(X \neg b) \rightarrow (\neg X b)$: true, the first one holds iff there is a discrete step and $\neg b$ holds in the next state, hence Xb is false.

$(\tilde{X} \neg b) \rightarrow (\neg \tilde{X} b)$:

$(G \tilde{X} \top) \rightarrow ((G b) \vee (G \neg b))$:

LTL- MTL Properties [1/2]

Let b be a boolean symbol. Are the following properties true or false?

$\tilde{Y}\top$: false in the initial state.

$(\neg Xb) \rightarrow (X\neg b)$: false, the first one holds in every time elapse, the second one holds only in discrete steps where $\neg b$ holds in the next state.

$(\neg\tilde{X}\ b) \rightarrow (\tilde{X}\neg b)$: false, as above but for time elapses.

$(X\neg b) \rightarrow (\neg Xb)$: true, the first one holds iff there is a discrete step and $\neg b$ holds in the next state, hence Xb is false.

$(\tilde{X}\neg b) \rightarrow (\neg\tilde{X}b)$: true, as above but for time elapses.

$(G\tilde{X}\top) \rightarrow ((Gb) \vee (G\neg b))$:

LTL- MTL Properties [1/2]

Let b be a boolean symbol. Are the following properties true or false?

$\tilde{Y}\top$: false in the initial state.

$(\neg Xb) \rightarrow (X\neg b)$: false, the first one holds in every time elapse, the second one holds only in discrete steps where $\neg b$ holds in the next state.

$(\neg\tilde{X}\ b) \rightarrow (\tilde{X}\neg b)$: false, as above but for time elapses.

$(X\neg b) \rightarrow (\neg Xb)$: true, the first one holds iff there is a discrete step and $\neg b$ holds in the next state, hence Xb is false.

$(\tilde{X}\neg b) \rightarrow (\neg\tilde{X}b)$: true, as above but for time elapses.

$(G\tilde{X}\top) \rightarrow ((Gb) \vee (G\neg b))$: true, the first part implies that we never perform a discrete transition and the truth value of b can only change in discrete transitions.

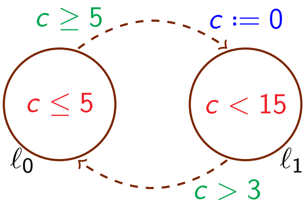
LTL- MTL Properties [2/2]

See files in examples.

Outline

Exercise 11.1: Simple Timed Automaton

Write the SMV model corresponding to the timed automaton in the figure.



Encode the following properties:

- from location ℓ_0 we always reach ℓ_1 within 5 time units
- if we are in ℓ_1 then for the next 3 time units we remain in ℓ_1
- if just arrived in ℓ_1 then for the next 3 time units we remain in ℓ_1.

Timed Thermostat

Exercise 11.2: Timed Thermostat

A thermostat has 2 states: on and off;

- if the temperature is below 18 degrees the thermostat switches on.
- if the temperature is above 18 degrees the thermostat switches off.

Every time the thermostat misures the temperature in the room, the temperature increases (if on) or decreases (if off) by dt (with respect to the previous check).

The thermostat measures the temperature at most ($<$) every max_dt time units.

The temperature initially is in $[18 - max\_dt, 18 + max\_dt]$.

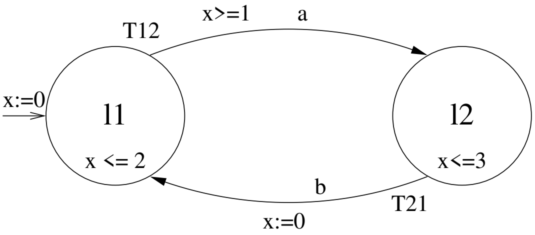
Verify that the temperature is always in $[18 - 2 \cdot max\_dt, 18 + 2 \cdot max\_dt]$

Outline

1. Timed nuXmv
2. Timed and Infinite Traces
3. Exercises
4. Homework

Homework 11.1: Timed Automata

Encode the timed automata in the figure below in nuXmv.

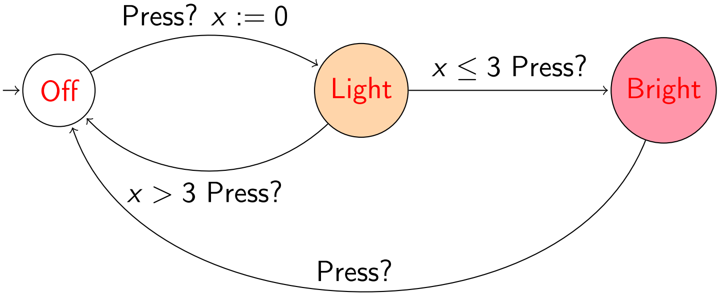


Check the following properties:

- From location ℓ_2 we always reach ℓ_1 within 1 time unit
- From location ℓ_2 we always reach ℓ_1 within 2 time units
- It is possible to be in location ℓ_1 at time $t = 1$

Homework 11.2: Light Control

Encode the timed automata in the figure below in nuXmv.



Verify the following properties:

- If the button is pressed infinitely often, then the light is turned on infinitely often.
- If the button is pressed infinitely often within 2 time units, then the light is bright infinitely often.

References

[1] Alessandro Cimatti et al. "Extending nuXmv with Timed Transition Systems and Timed Temporal Properties". In: *CAV 2019.*