



UNIVERSITÀ DI TRENTO

Formal Method Mod. 2 (Model Checking)

Laboratory 12

Giuseppe Spallitta
giuseppe.spallitta@unitn.it

Università degli studi di Trento

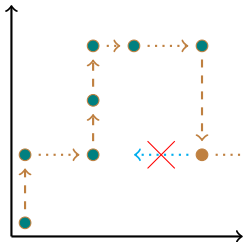
June 5, 2024

Hybrid systems

Cyber-physical systems

- ▶ Discrete controller with some modes: (in)finite state automaton; e.g. electronic controller.
 - ▶ continuous variables with some behaviour w.r.t. time, physical phenomena e.g. braking car, water pump, temperature.
-
- ▶ in general model checking on hybrid systems is **undecidable**.
 - ▶ many sub-classes
 - decidable: *rectangular*, *singular*;
 - undecidable: *linear*;

discrete

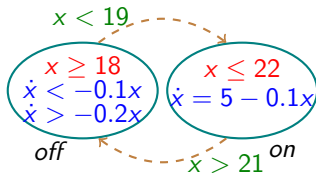


time

Hybrid systems: representation

Hybrid Automata

- Explicit graph representation of discrete states/modes (nodes) and transitions (edges);
- Symbolic representation of linear temporal aspects via polytopes (N dimensional polyhedron);
- **location invariants**,
- **transition guards**,
- **flow**: derivative w.r.t time.



Outline

1. HyComp
2. Example
3. Homework



HyComp introduction

- ▶ HyComp has been developed in Embedded Systems (FBK) as part of Sergio Mover's PhD.
- ▶ Supports the modelling and verification of a network of hybrid automata;
- ▶ Supports invariant and LTL properties;
- ▶ It encodes the Hybrid model into a “standard” nuXmv model.

HyComp: input language [1/3]

The input language of HyComp is called HyDi (Hybrid automata with Discrete interaction).

Overview

- ▶ `main` module contains description of the network of automata: processes are `MODULE` instances;
- ▶ `main` module contains synchronization constraints: `EVENT`;
- ▶ Symbolic description of infinite transition system using: `INIT`, `INVAR` and `TRANS` to specify initial, invariant and transition conditions.
- ▶ `continuous` type variables with `FLOW` conditions,

HyComp: input language [2/3]

HyComp adds

- ▶ continuous variable type;
- ▶ all continuous vars increase accordingly to their `FLOW` conditions in timed transitions;
- ▶ time: built-in continuous symbol with flow condition: `der(time) = 1`, can not be used in properties;
- ▶ URGENT: freeze time: when one of the URGENT conditions is satisfied only discrete transitions are allowed;



HyComp: input language [3/3]

HyComp updates

- ▶ TRANS constrain the discrete behaviour only,
- ▶ INVAR: continuous allowed in invariants with shape:
`no_continuous_expr -> convex_continuous_expr.`



HyComp: commands

read and rewrite model

1. `hycomp_read_model`
2. `hycomp_compile_model`
3. `hycomp_untime_network`
4. `hycomp_async2sync_network`
5. `hycomp_net2mono`

check specifications

- ▶ `hycomp_check_invar_*`
- ▶ `hycomp_check_ltl*`

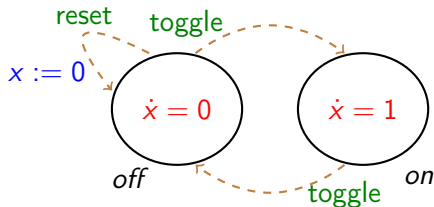
Outline

1. HyComp
2. Example
3. Homework



Example: stopwatch [1/3]

- ▶ write a HyDi model that represents the hybrid automaton in the picture.
- ▶ add an asynchronous process that controls the stop-watch using the `toggle` and `reset` commands.



Example: stopwatch [2/3]

Stopwatch module

```
MODULE Stopwatch
  DEFINE
    on  := mode = _on;
    off := mode = _off;

  VAR
    mode : {_on, _off};
    c : continuous;

  EVENT toggle, reset;
  FLOW on -> der(c) = 1;
  FLOW off -> der(c) = 0;

  TRANS EVENT = reset -> next(c) = 0;
  TRANS EVENT != reset -> next(c) = c;
  TRANS EVENT = toggle -> next(mode) != mode;
```

Example: stopwatch [3/3]

Controller module

```
MODULE Controller
EVENT toggle, reset;
```

main module

```
MODULE main
VAR
    stopWatch : Stopwatch;
    controller : Controller;

SYNC controller, stopWatch EVENTS toggle, toggle;
SYNC controller, stopWatch EVENTS reset, reset;
```

Outline

1. HyComp
2. Example
3. Homework



Bouncing ball

- ▶ A ball is initially at height 10.
- ▶ We let the ball fall and bounce.
- ▶ Every time the ball bounces half its speed is lost.
- ▶ The gravitational acceleration is 9.8.

