Proving and Explaining the Unfeasibility of Message Sequence Charts for Hybrid Systems

Alessandro Cimatti **Sergio Mover** Stefano Tonetta

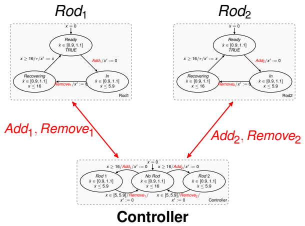
Fondazione Bruno Kessler

October 31, 2011

Motivations

Hybrid Systems

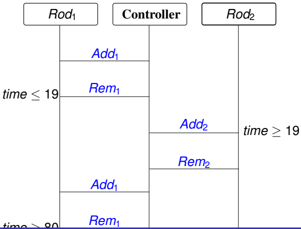
- Mix discrete (e.g. hardware) and continuous (e.g. sensor) behaviors.
- Complex critical systems: train control system (ETCS), airplane traffic control system (TCAS), . . .
- Network of components.



Scenario-verification

Is there a run of the system compatible with the scenario?

If such a run exists, the scenario is feasible.



Motivations

Existing approaches:

1. Reduction to reachability:
   - Can prove both feasibility and unfeasibility.
   - Inefficient.
2. Scenario-based encoding [CAV11]:
   - Cannot prove unfeasibility.
   - Efficient.

Our contribution is a SMT-based technique that:

- Efficiently proves unfeasibility.
- Extracts explanations for the unfeasibility.

Outline

Outline

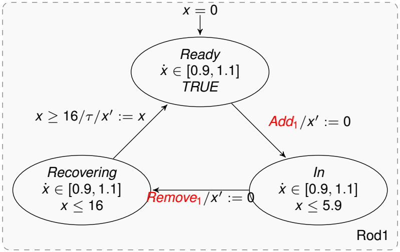
Outline

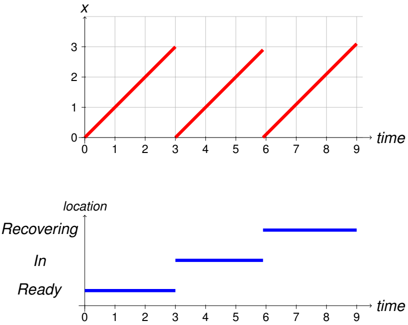
1. Background
   - SMT analysis of Hybrid Systems
   - Scenario-Verification
2. Proving the unfeasibility of scenarios
3. Explanations of Unfeasibility
4. Experimental Evaluation
5. Conclusions and future work

Hybrid Automata

Hybrid automata ([Henzinger 96]):

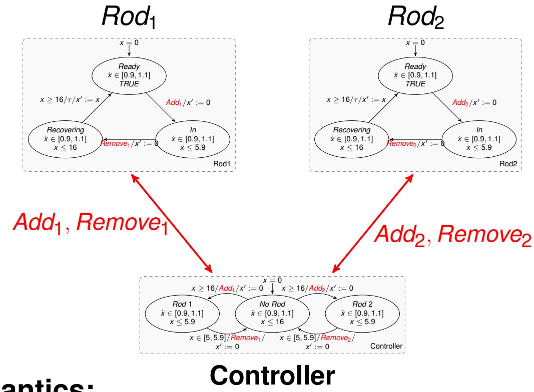
- Framework for representing hybrid systems.
- Discrete instantaneous mode switches.
- Continuous evolution according to flow conditions.





Network of hybrid automata $\mathcal{H} = H_1 || \ldots || H_n$:

- Move asynchronously on local events (τ).
- Synchronize on shared events.

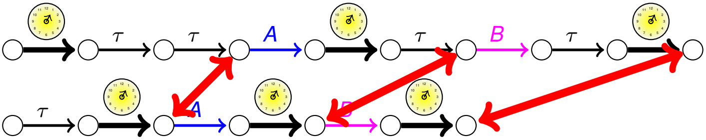


Different semantics:

1. Global-time ([Henzinger 96]).
2. **Local-time** ([Bengstsson 98]).

Local-time semantics

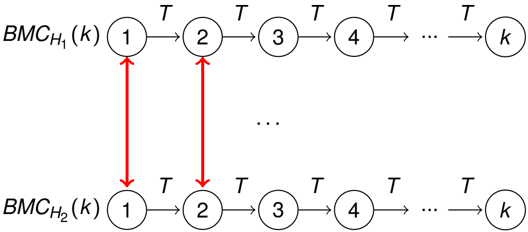
- The time evolves independently in each automaton:
  - Local time scale.
  - The continuous evolution is a local transition.
- The local time of the automata must be the same:
  - On synchronizations.
  - At the end of a run.



τ = local event (no stutter or time).

SMT analysis of Hybrid Systems

- Each automaton is encoded in a symbolic transition system $H_i = \langle Init_i, Trans_i \rangle$.
- Bounded model checking:



- k-induction.
  - Base case: BMC up to k.
  - Inductive case: BMC and simple path condition up to $k + 1$.
- Use SMT solvers as decision procedure.

Outline

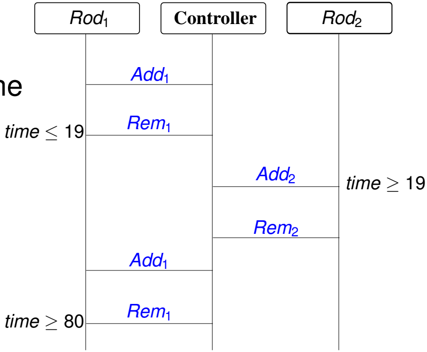
Constrained Message Sequence Charts

$\langle m, \phi \rangle$: Message sequence chart m with constraints ϕ.

m: parallel composition of instances.

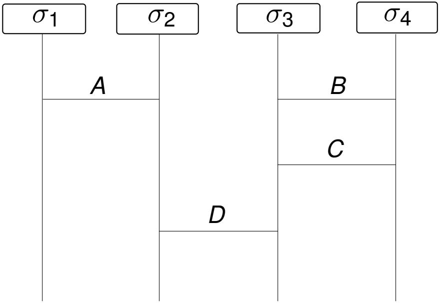
$\phi = \phi_g \wedge \phi_1 \wedge \ldots \wedge \phi_n$: formulas over the network variables on synchronization.

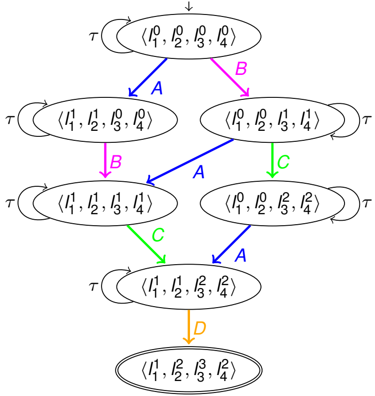
- Global (ϕ_g): over all the network variables.
- Local ϕ_i: over variable of H_i.



MSC verification via reachability

- The CMSC is translated in a monitor automaton S_m.
- The automaton is composed with the network.
- Enables off-the-shelf verification techniques:
  - BMC: feasibility.
  - k-induction: unfeasibility.

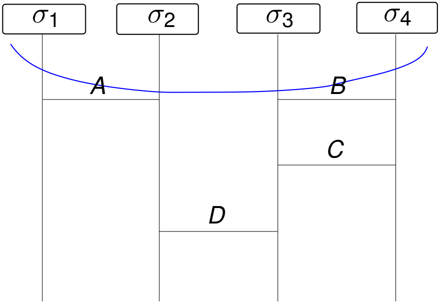


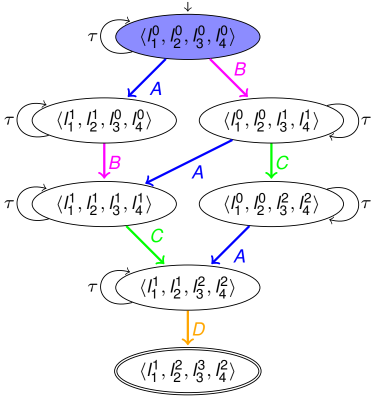


$$m = \sigma_1 || \sigma_2 || \sigma_3 || \sigma_4$$

MSC verification via reachability

- The CMSC is translated in a monitor automaton S_m.
- The automaton is composed with the network.
- Enables off-the-shelf verification techniques:
  - BMC: feasibility.
  - k-induction: unfeasibility.

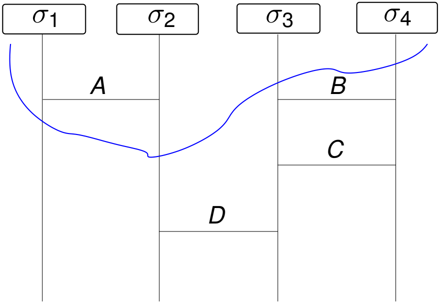


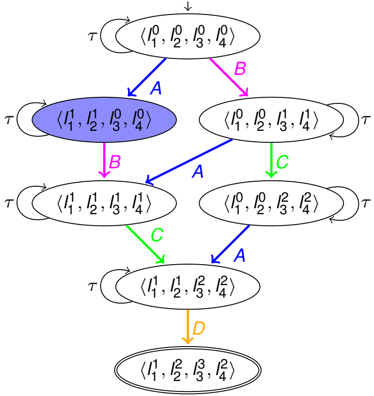


Cut: $\langle l_1^0, l_2^0, l_3^0, l_4^0 \rangle$

MSC verification via reachability

- The CMSC is translated in a monitor automaton S_m.
- The automaton is composed with the network.
- Enables off-the-shelf verification techniques:
  - BMC: feasibility.
  - k-induction: unfeasibility.

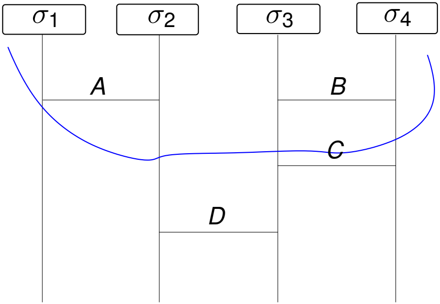


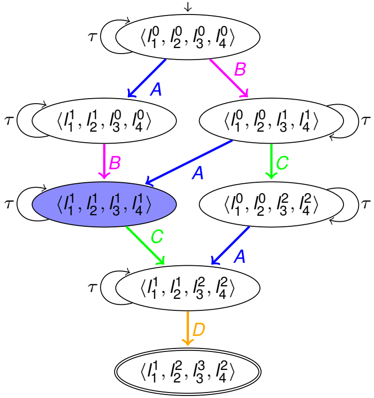


Cut: $\langle l_1^1, l_2^1, l_3^0, l_4^0 \rangle$

MSC verification via reachability

- The CMSC is translated in a monitor automaton S_m.
- The automaton is composed with the network.
- Enables off-the-shelf verification techniques:
  - BMC: feasibility.
  - k-induction: unfeasibility.

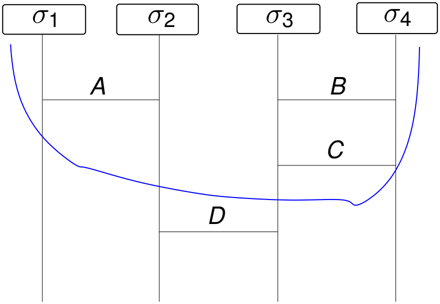




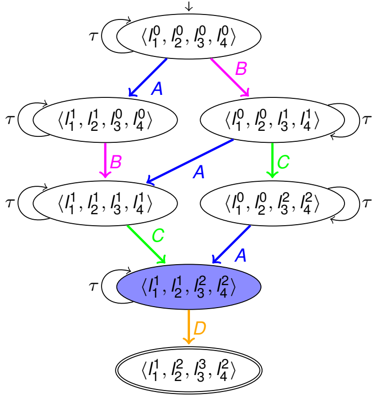
Cut: $\langle l_1^1, l_2^1, l_3^1, l_4^1 \rangle$

MSC verification via reachability

- The CMSC is translated in a monitor automaton S_m.
- The automaton is composed with the network.
- Enables off-the-shelf verification techniques:
  - BMC: feasibility.
  - k-induction: unfeasibility.

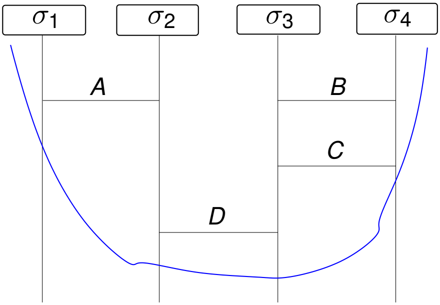


Cut: $\langle l_1^1, l_2^1, l_3^2, l_4^2 \rangle$

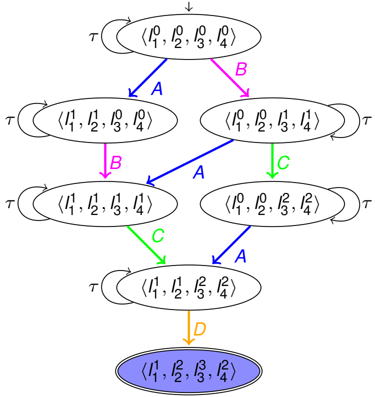


MSC verification via reachability

- The CMSC is translated in a monitor automaton S_m.
- The automaton is composed with the network.
- Enables off-the-shelf verification techniques:
  - BMC: feasibility.
  - k-induction: unfeasibility.

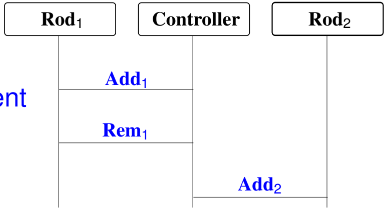


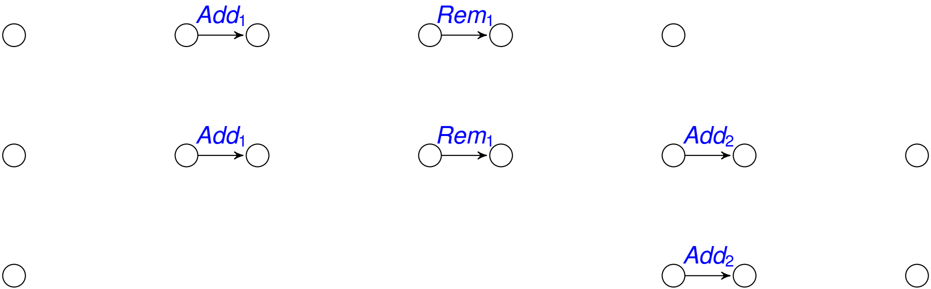
Cut: $\langle l_1^1, l_2^3, l_3^3, l_4^2 \rangle$



Scenario-based encoding

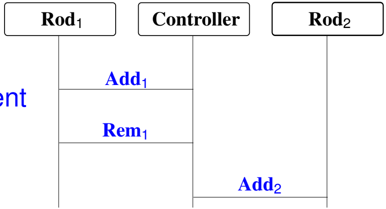
- For all the automata:
  - Fix the position of the shared events.
    transitions are simplified wrt shared event

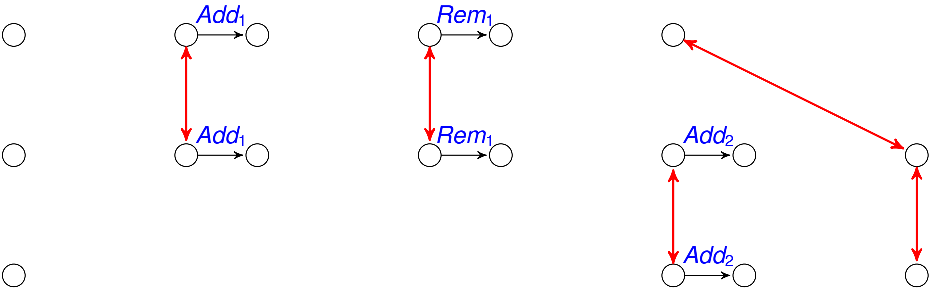




Scenario-based encoding

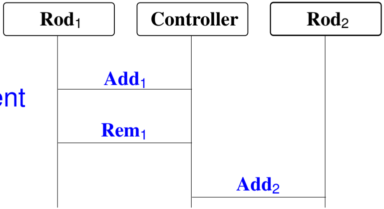
- For all the automata:
  - Fix the position of the shared events.
    transitions are simplified wrt shared event
  - Add the synchronization constraints.

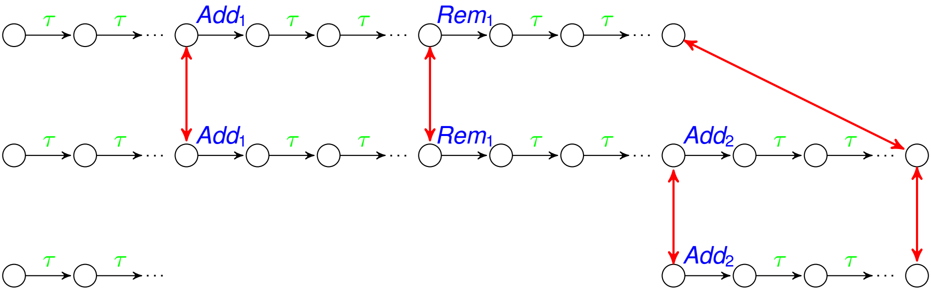




Scenario-based encoding

- For all the automata:
  - Fix the position of the shared events.
    transitions are simplified wrt shared event
  - Add the synchronization constraints.
  - Encode the "local segments".
    transitions are simplified wrt τ





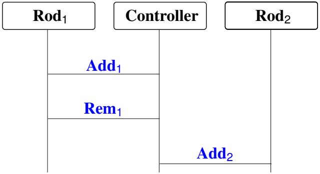
Outline

Efficient unfeasibility check

	Reduction to reachability	SMT-based approach
Feasibility	BMC Inefficient	Scenario-driven encoding Efficient
Unfeasibility	K-induction Inefficient	Partitioned k-induction Efficient

Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.



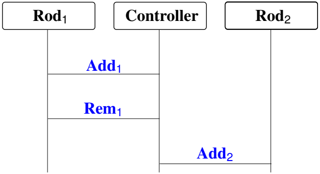
○

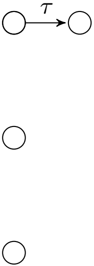
○

○

Partitioned K-induction - Algorithm

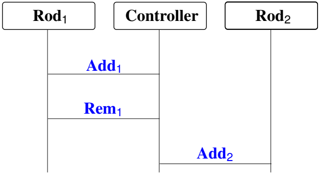
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

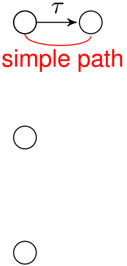




Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

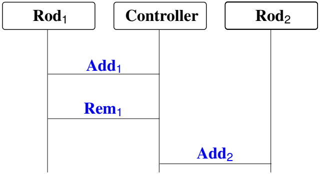


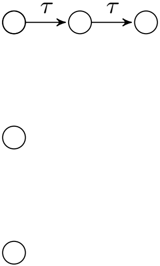


SAT - new states are reachable

Partitioned K-induction - Algorithm

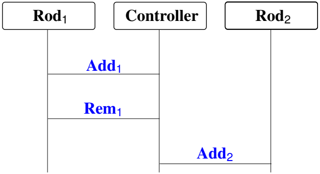
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

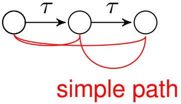




Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

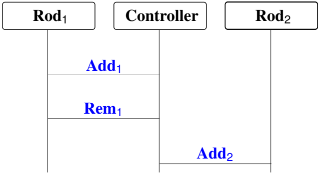


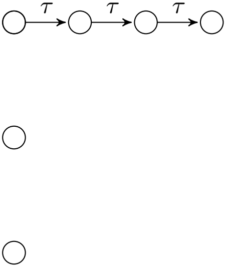


SAT - new states are reachable

Partitioned K-induction - Algorithm

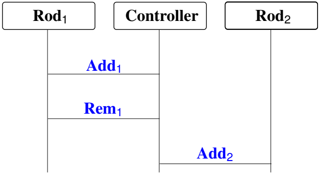
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

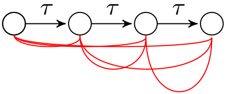




Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.



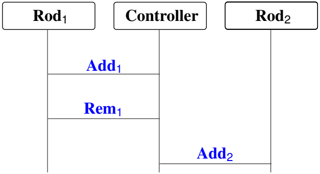


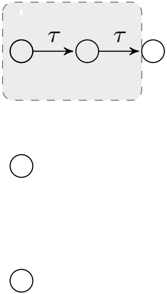
UNSAT - no new states are reachable

simple path

Partitioned K-induction - Algorithm

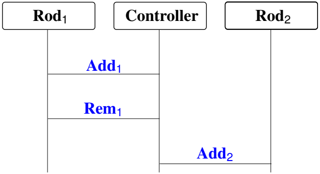
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

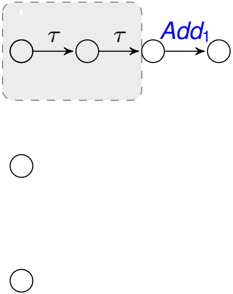




Partitioned K-induction - Algorithm

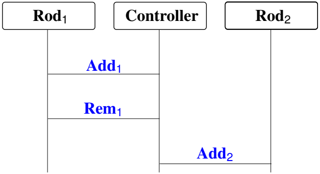
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

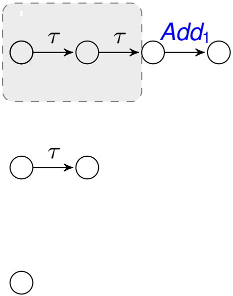




Partitioned K-induction - Algorithm

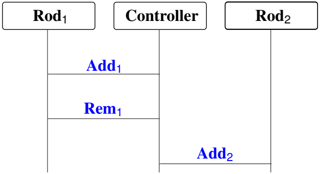
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

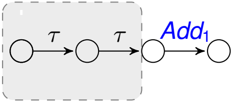


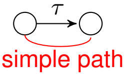


Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.



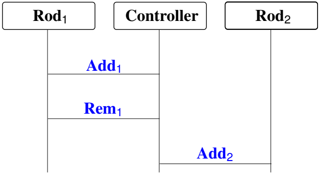


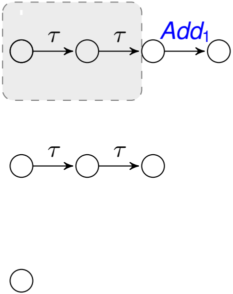


SAT - new states are reachable

Partitioned K-induction - Algorithm

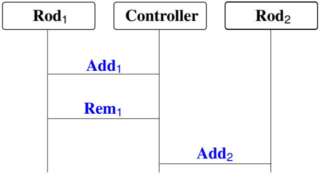
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

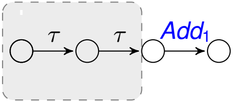


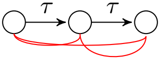


Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.





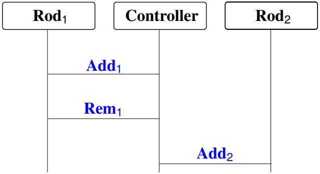


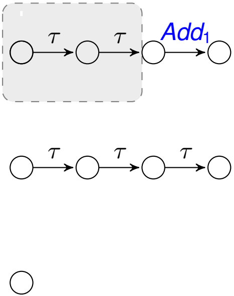
SAT - new states are reachable

simple path

Partitioned K-induction - Algorithm

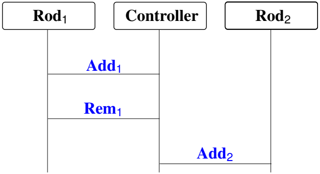
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

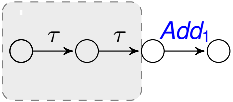


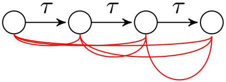


Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.



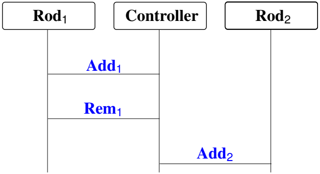


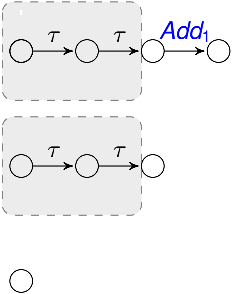


simple path

Partitioned K-induction - Algorithm

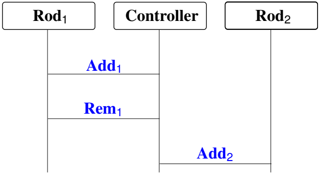
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

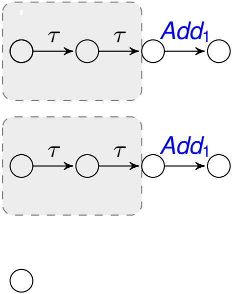




Partitioned K-induction - Algorithm

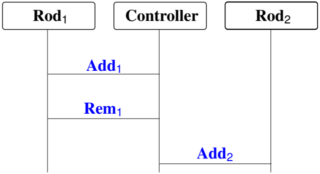
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

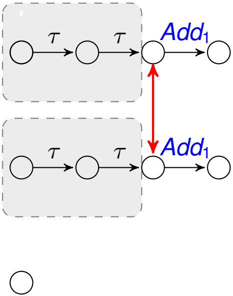




Partitioned K-induction - Algorithm

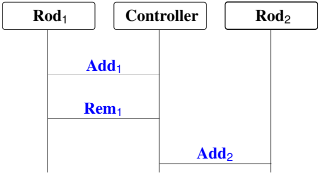
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

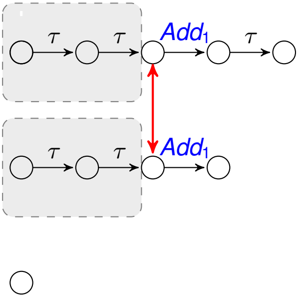




Partitioned K-induction - Algorithm

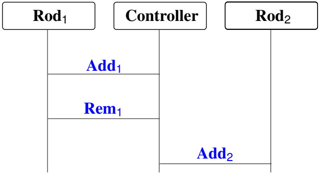
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

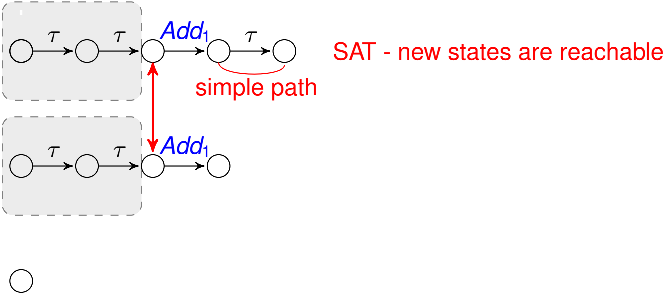




Partitioned K-induction - Algorithm

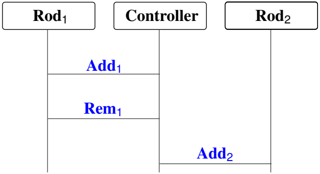
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

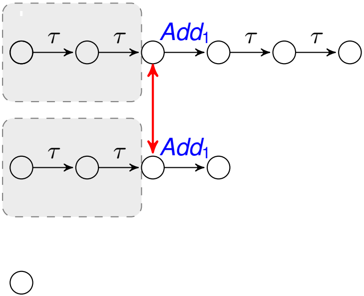




Partitioned K-induction - Algorithm

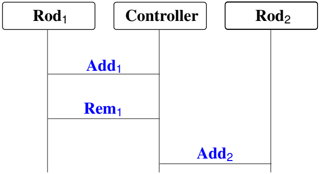
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

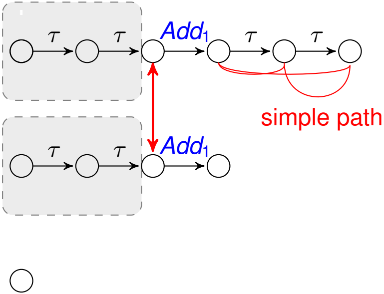




Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

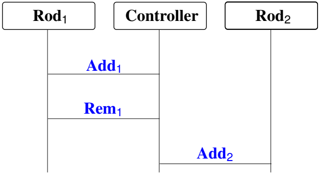


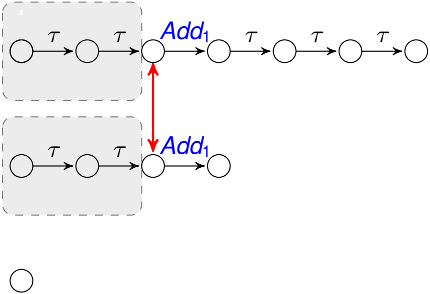


SAT - new states are reachable

Partitioned K-induction - Algorithm

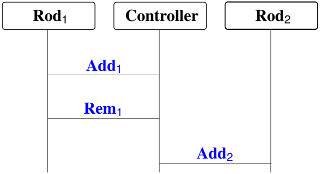
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

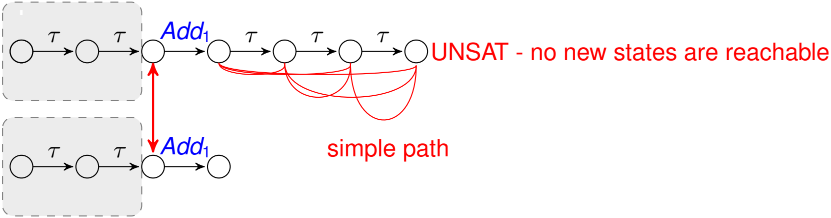




Partitioned K-induction - Algorithm

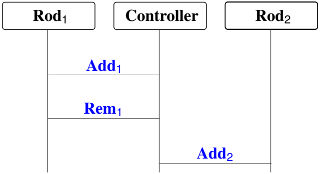
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

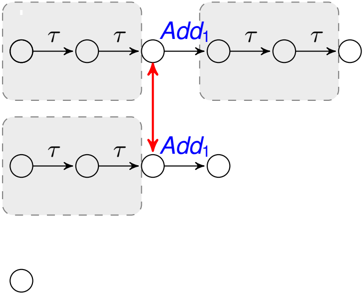




Partitioned K-induction - Algorithm

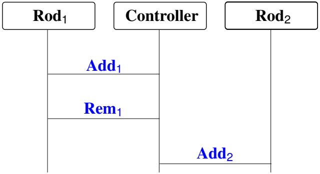
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

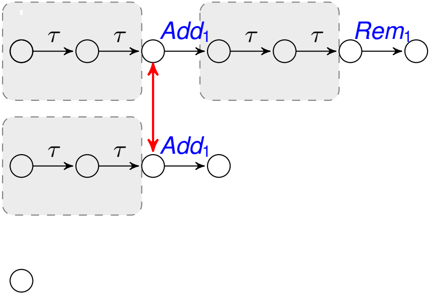




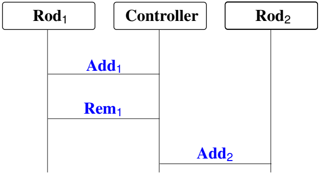
Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

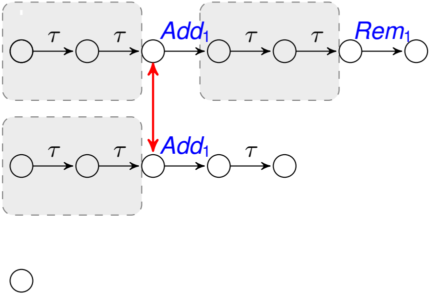




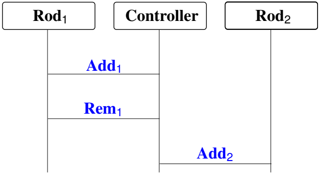
Partitioned K-induction - Algorithm

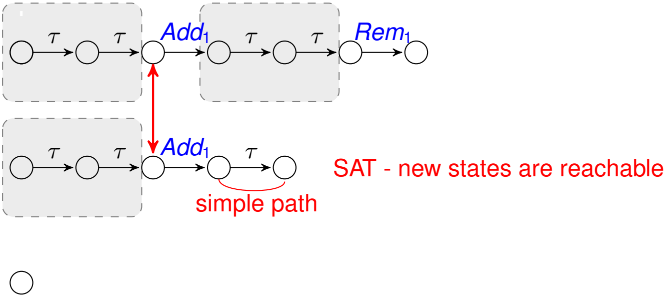


- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.



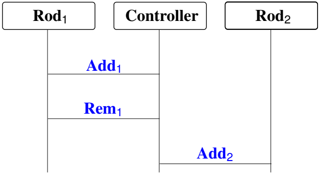
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

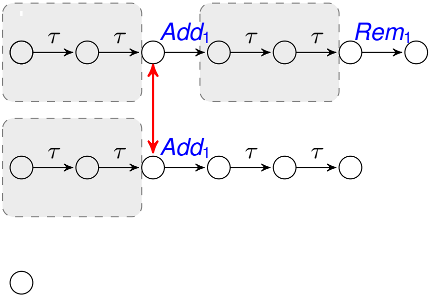




Partitioned K-induction - Algorithm

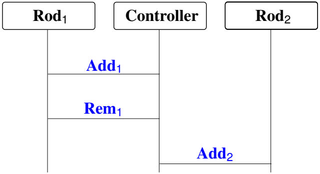
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

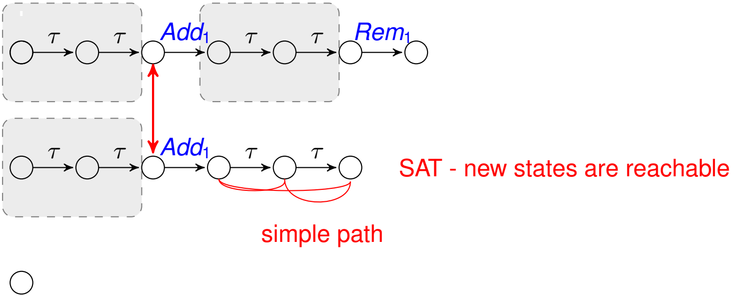




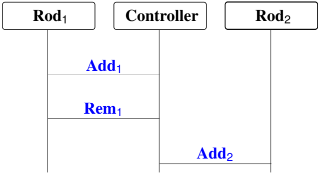
Partitioned K-induction - Algorithm

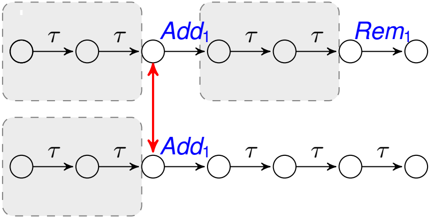
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.





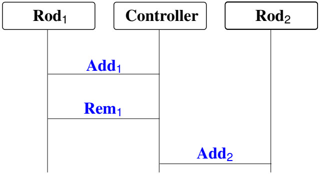
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

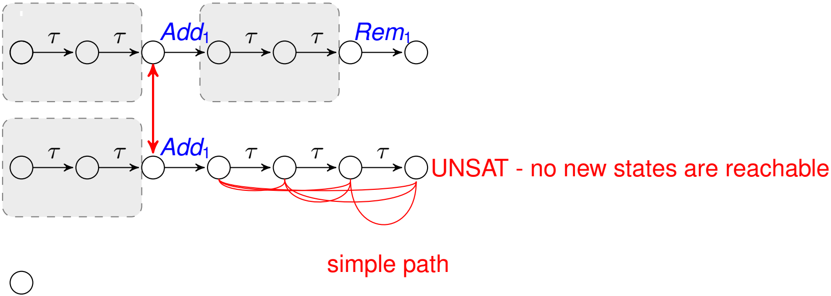




Partitioned K-induction - Algorithm

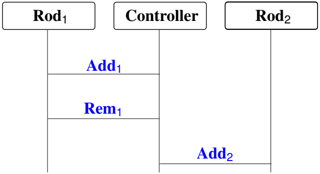
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

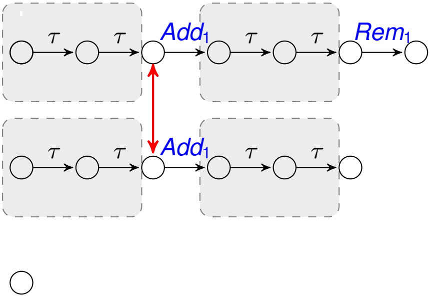




Partitioned K-induction - Algorithm

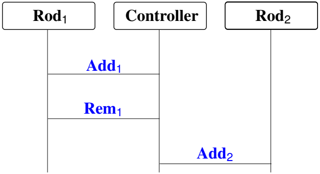
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

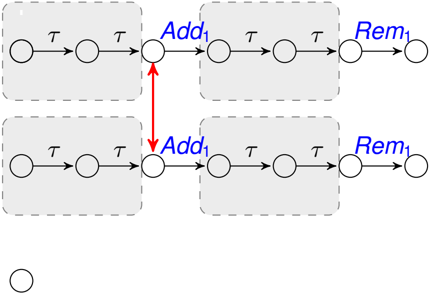




Partitioned K-induction - Algorithm

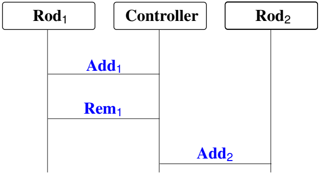
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

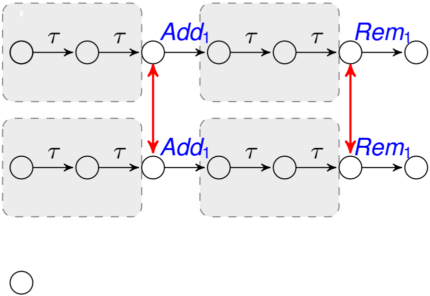




Partitioned K-induction - Algorithm

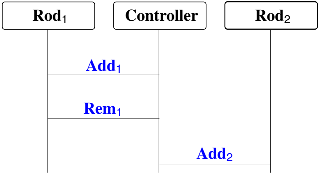
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

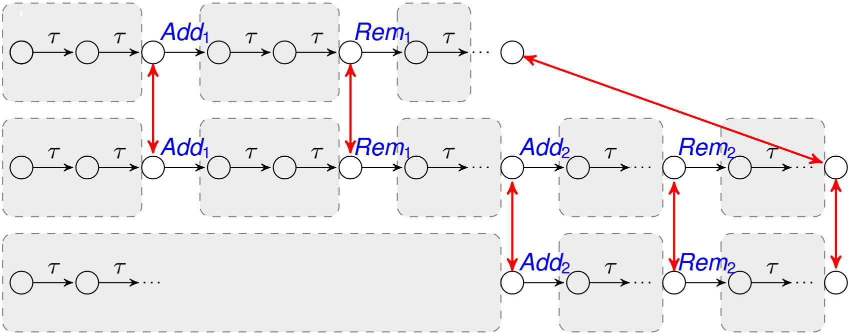




Partitioned K-induction - Algorithm

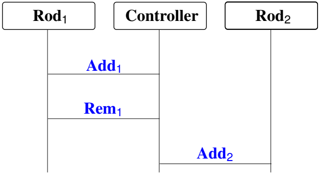
- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.

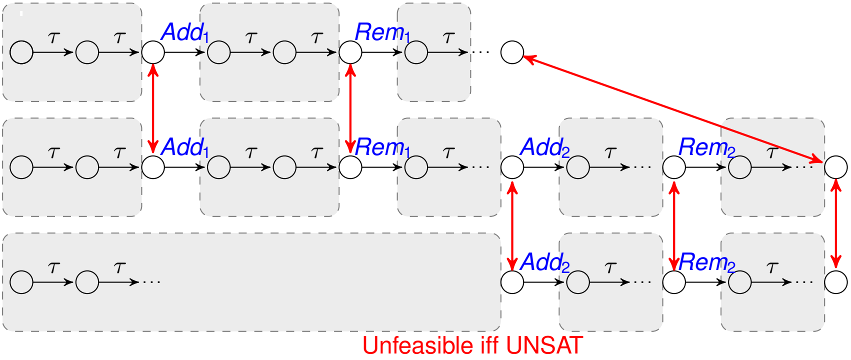




Partitioned K-induction - Algorithm

- Inductive step: proved incrementally following the partial order of the MSC.
- Base case: bounded feasibility check.





Outline

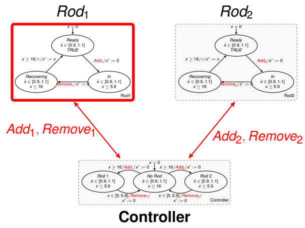
Explanations of unfeasibility

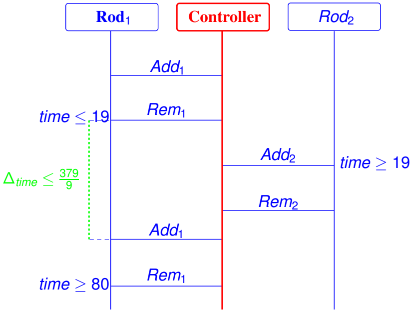
- Typical use case:
  - We expect that a scenario is feasible.
  - The analysis proves that the scenario is unfeasible in the network.
  - How do we explain the unfeasibility?
- We extract three types of explanations for the unfeasibility.

Unfeasibility due to a component

Explained with a formula that:

- Is required by the component when simulating its MSC events.
- Is not consistent with the other components when they simulate the events of the MSC.





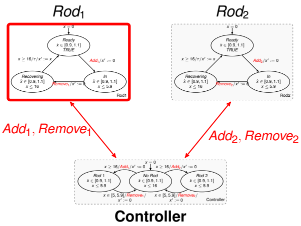
Unfeasibility due to a component

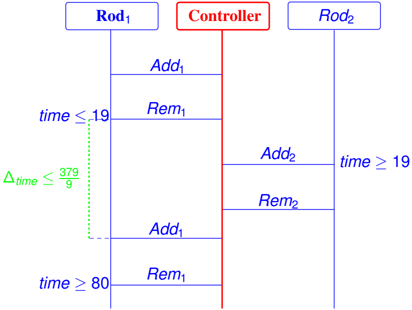
Explained with a formula that:

- Is required by the component when simulating its MSC events.
- Is not consistent with the other components when they simulate the events of the MSC.

It is the interpolant of A and B:

- A is the encoding of the component and its MSC events.
- B is the encoding of the other components and their MSC events.

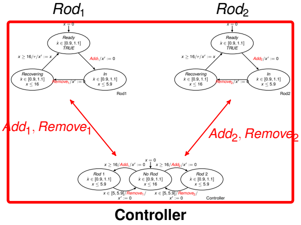


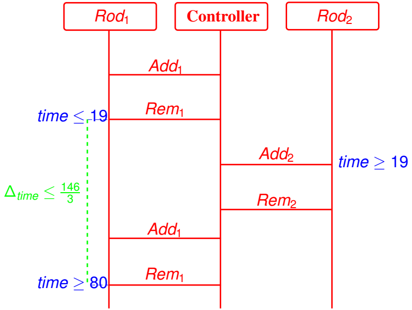


Unfeasibility due the network

Explained with a formula that:

- Is required by the network when simulating the MSC.
- Is not consistent with the additional constraints of the MSC.



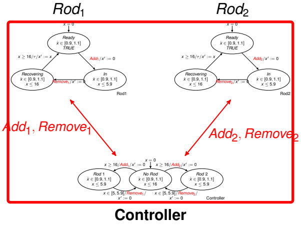


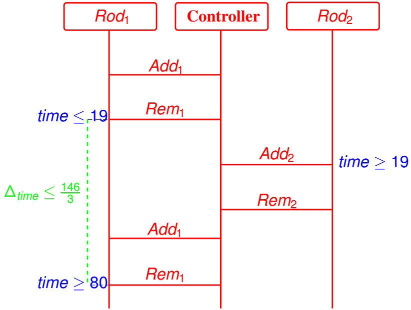
Explained with a formula that:

- Is required by the network when simulating the MSC.
- Is not consistent with the additional constraints of the MSC.

It is the interpolant of A and B:

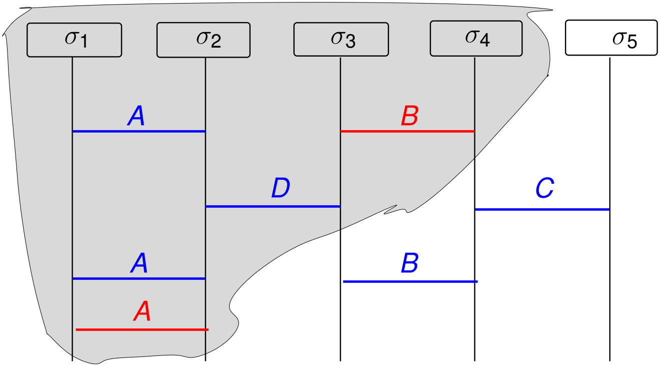
- A is the encoding of the network and the MSC.
- B are the CMSC constraints.





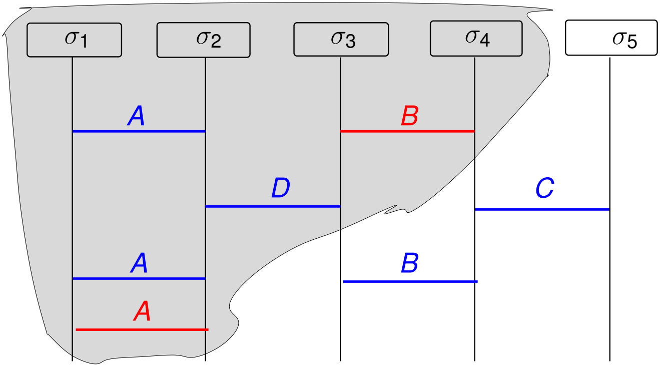
Inconsistent subset of the CMSC

Subset of the original CMSC that is still unfeasible with the network.



Inconsistent subset of the CMSC

Subset of the original CMSC that is still unfeasible with the network.
Extracted from the unsatisfiable core of the encoding.



Outline

Experimental Evaluation

Implementation:

- Approach implemented on top of the NuSMV model checker.
- We use the MathSAT SMT solver.

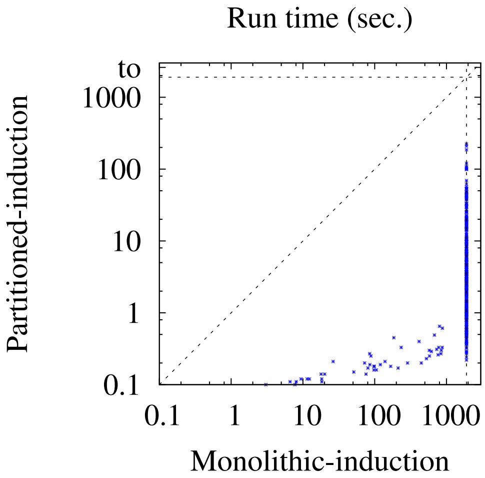
Settings:

- Linear hybrid automata benchmarks.
- Several handcrafted (unsatisfiable) MSCs.
- We scaled the dimension of the benchmarks (number of automata, length of the MSCs).

Comparison:

- MSC partitioned k-induction.
- Monolithic k-induction on the system composed with the monitor automata.

Partitioned k-induction vs. Monolithic k-induction (run times)



Outline

Conclusions and future work

- Efficient approach for proving the unfeasibility of CMSC.
  - The encoding exploits the structure of the CMSC.
  - Partitioned k-induction.
- Unfeasibility explanations:
  - Useful to localize and correct the errors.
  - Extracted exploiting the SMT solver functionalities.

Future works:

- More expressive MSCs (e.g. partial MSCs specifications).
- Validate the extracted explanations by real users.
- Automatic refinement loop in the abstraction.
- Non-linear hybrid systems.

Thank you for your attention.