Nantes Université

cnrs

LS2N Laboratoire des Sciences du Numérique de Nantes

CSL 2026

Paris, France

26th February 2026

Parametric disjunctive timed networks

Étienne André, Swen Jacobs, Engel Lefaucheux

LS2N, Nantes Université, Bretagne, France
CISPA Helmholtz Center for Information Security, Germany
Inria, LORIA, Nancy, France

Nantes Université

cnrs

LS2N Laboratoire des Sciences du Numérique de Nantes

CSL 2026

Paris, France

26th February 2026

Parametric disjunctive timed networks

(in one slide)

Étienne André, Swen Jacobs, Engel Lefaucheux

LS2N, Nantes Université, Bretagne, France
CISPA Helmholtz Center for Information Security, Germany
Inria, LORIA, Nancy, France

Concurrent multi-agent real-time systems

Complex systems involving

- Multiple agents
- Concurrency
- Real-time constraints

Hard!

Concurrent multi-agent real-time systems

Complex systems involving

- Multiple agents
- Concurrency
- Real-time constraints

Hard!

+ uncertainty

? unknown number of agents

? timed uncertainty (delays known with finite precision, or unknown)

Even harder!

Model checking in a nutshell

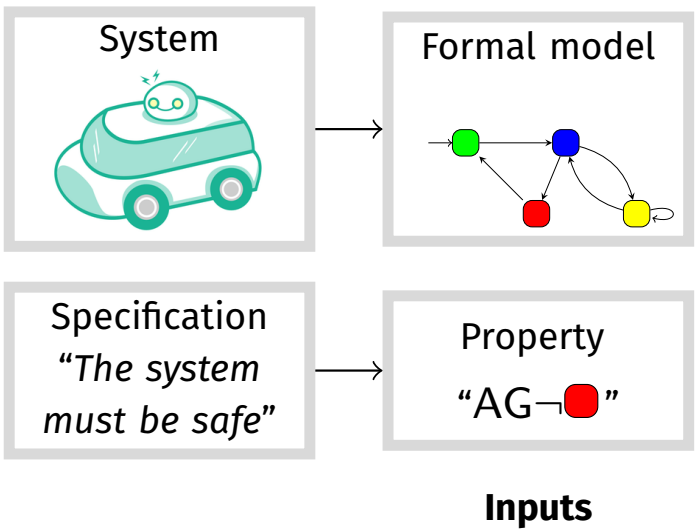
- Principle of model checking

System

Specification
"The system must be safe"

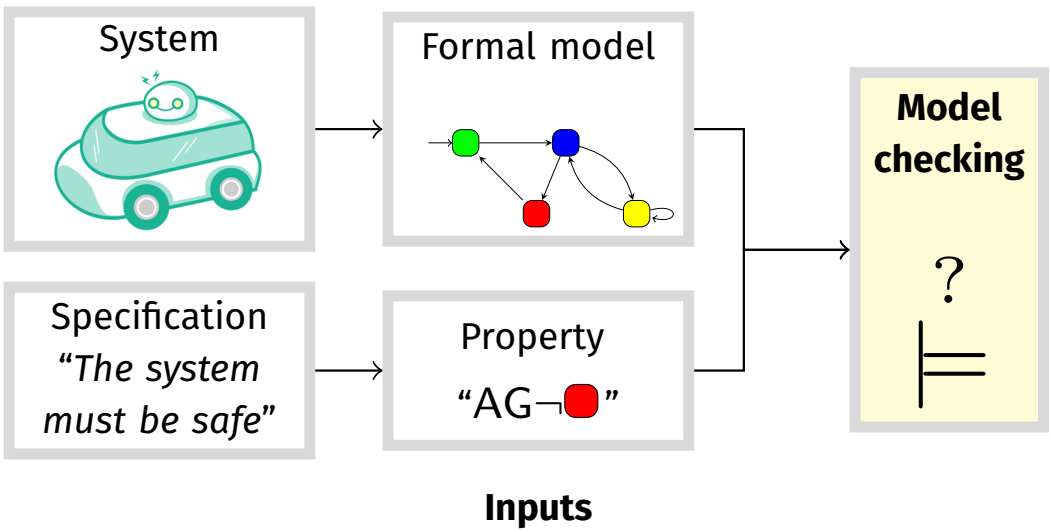
Model checking in a nutshell

- Principle of model checking



Model checking in a nutshell

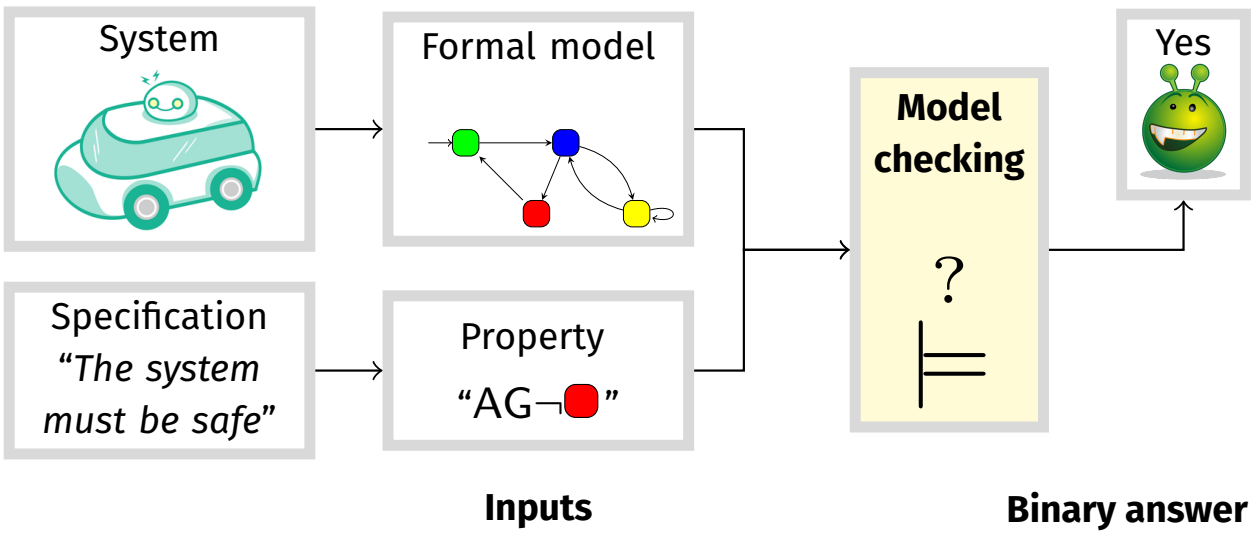
- Principle of model checking



- Question: does the model of the system satisfy the property?

Model checking in a nutshell

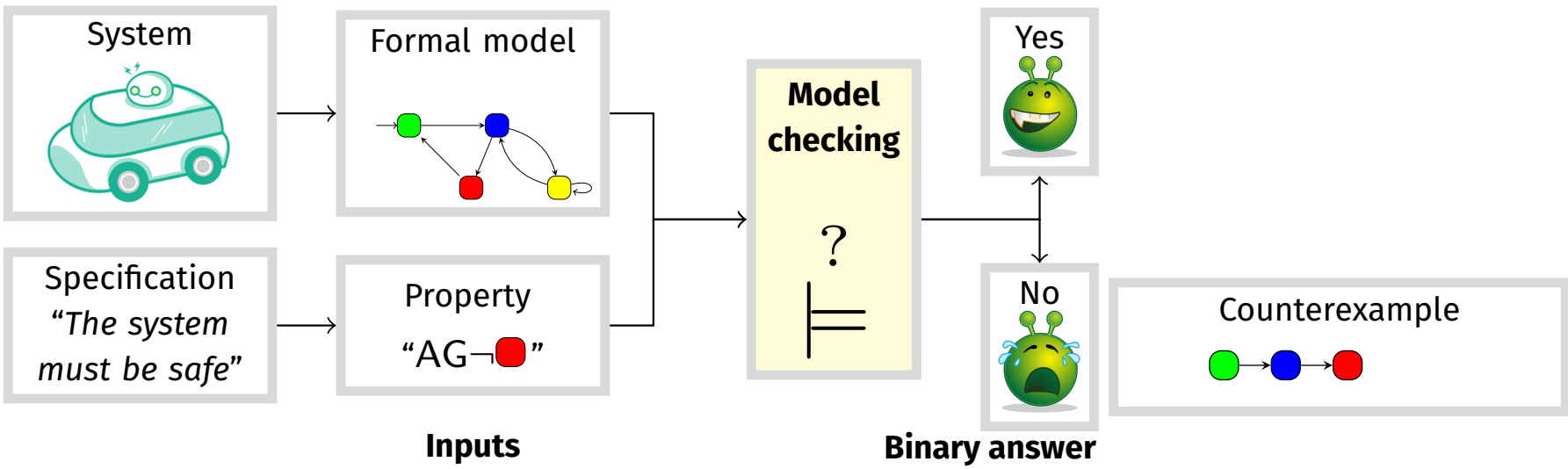
- Principle of model checking



- Question: does the model of the system satisfy the property?

Model checking in a nutshell

- Principle of model checking



- Question: does the model of the system satisfy the property?

Outline

Timed automata

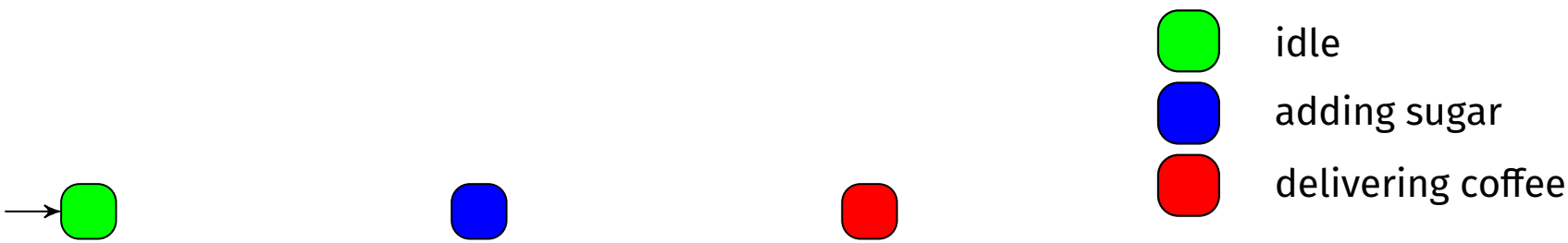
Extension of finite-state automata to concurrency [AD94]

- ☺ multiple agents (multiple timed automata in parallel)
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ a number of decidable problems

[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed automaton (TA)

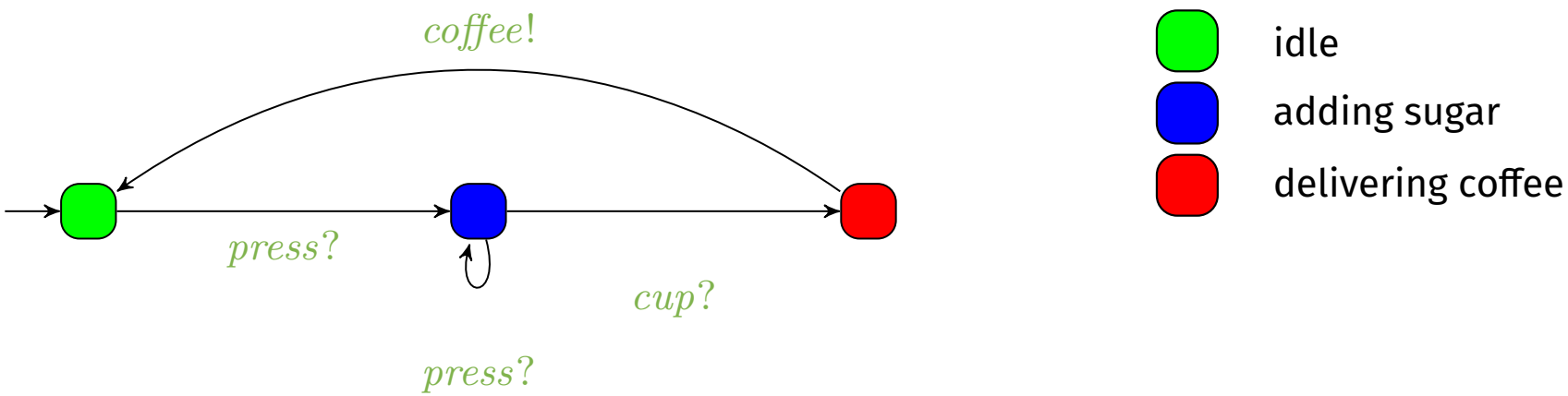
- Finite-state automaton (sets of locations)



[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed automaton (TA)

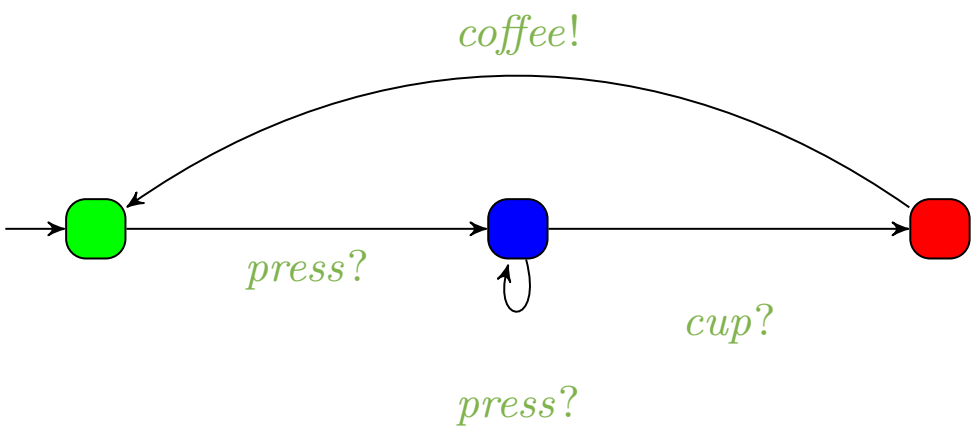
- Finite-state automaton (sets of locations and actions)

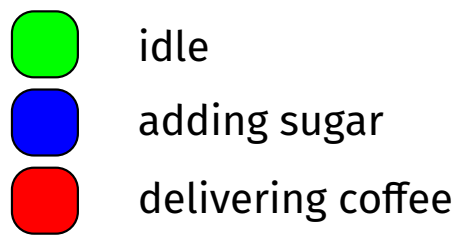


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed automaton (TA)

- Finite-state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate



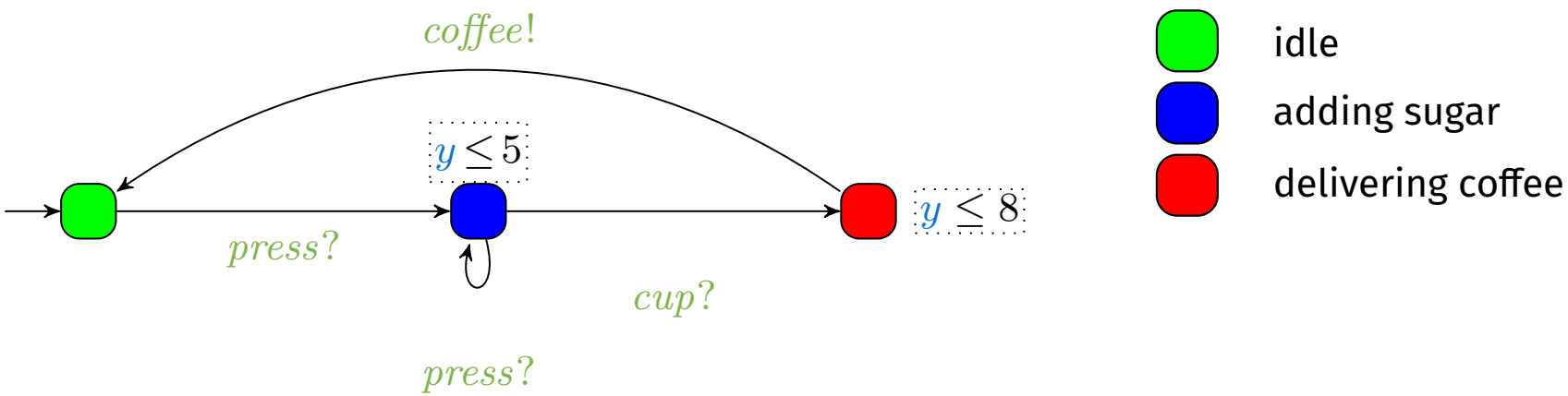


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed automaton (TA)

- Finite-state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants

- Features
  - Location invariant: property to be verified to stay at a location

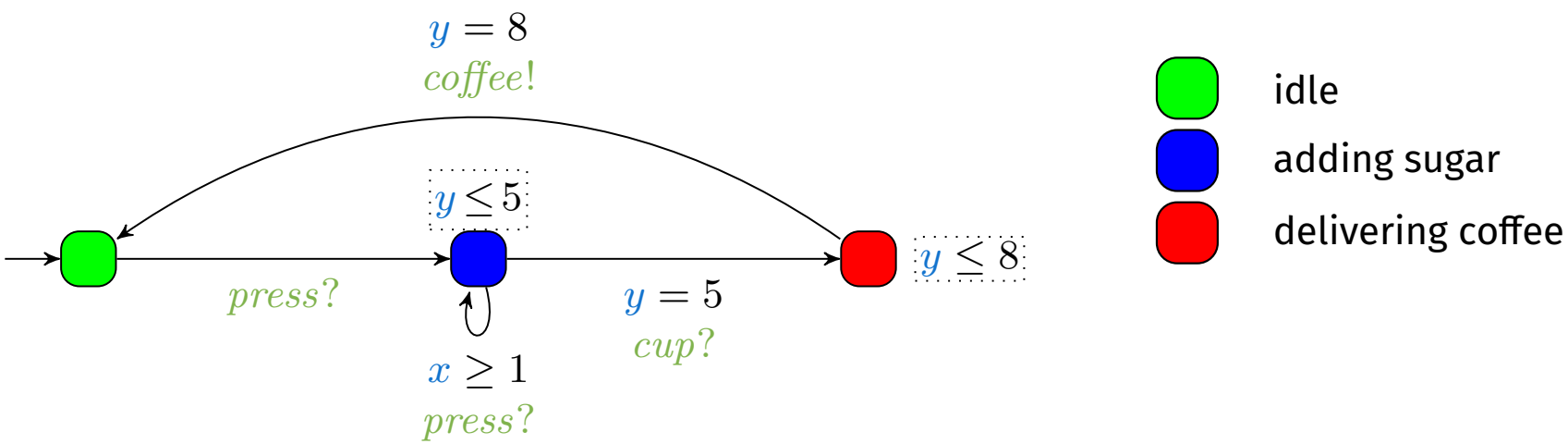


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed automaton (TA)

- Finite-state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards

- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition

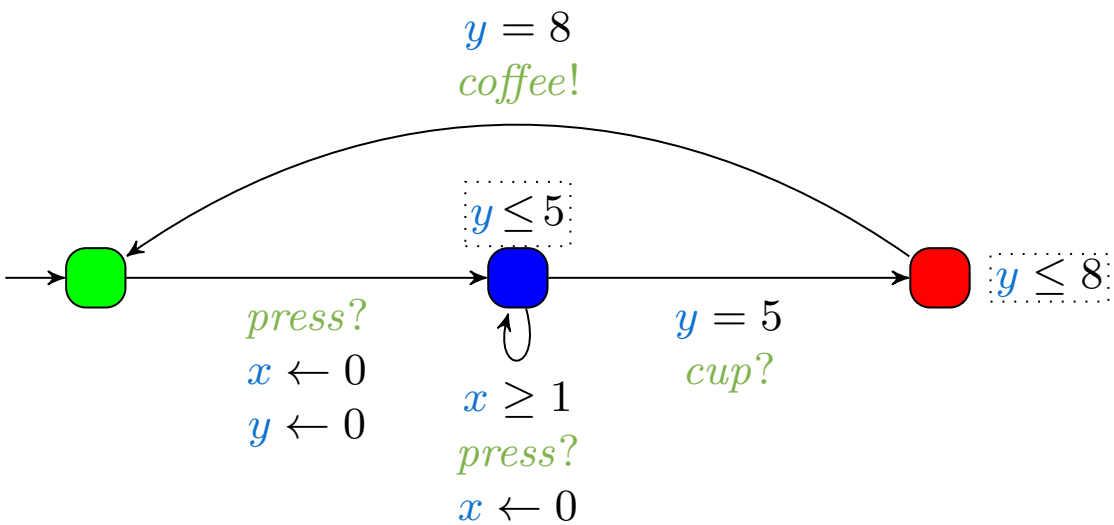


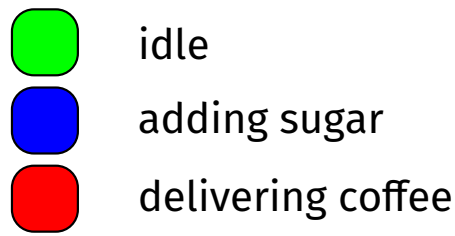
[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed automaton (TA)

- Finite-state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards

- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition
  - Clock reset: some of the clocks can be set to 0 along transitions





[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed automata: advantages and drawbacks

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ a number of decidable problems

Timed automata: advantages and drawbacks

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ a number of decidable problems
- ☹ fixed and *a priori* known number of agents

Timed automata: advantages and drawbacks

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ a number of decidable problems
- ☹ fixed and *a priori* known number of agents
- ☹ requires infinite precision w.r.t. time

Outline

Outline

Need for timed uncertainty

- Challenge 1: systems incompletely specified
  - Some delays may not be known yet, or may change
- Challenge 2: Robustness [Mar11]
  - What happens if 8 is implemented with 7.99?
- Challenge 3: Optimization of timing constants
- Challenge 4: Avoiding numerous verifications
  - If one of the timing delays of the model changes, should I model check again the whole system?

[Mar11] Nicolas Markey. "Robustness in Real-time Systems". In: *SIES*. IEEE Computer Society Press, June 2011, pp. 28–34

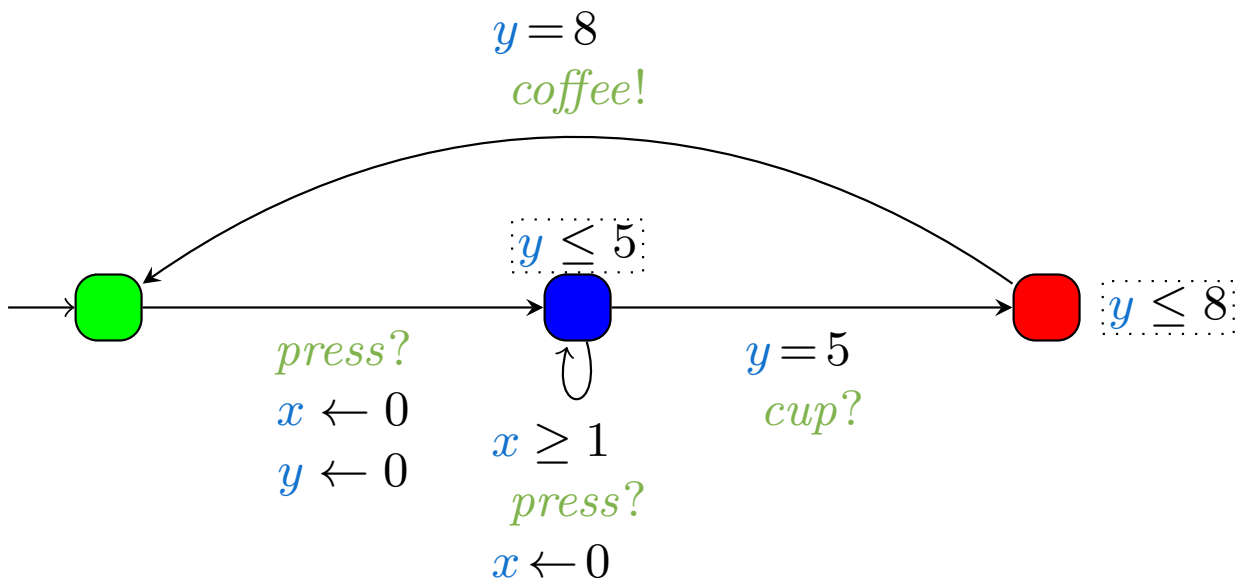
Need for timed uncertainty

- Challenge 1: systems incompletely specified
  - Some delays may not be known yet, or may change
- Challenge 2: Robustness [Mar11]
  - What happens if 8 is implemented with 7.99?
- Challenge 3: Optimization of timing constants
- Challenge 4: Avoiding numerous verifications
  - If one of the timing delays of the model changes, should I model check again the whole system?
- A solution: Parametric analysis
  - Consider that timing constants are unknown (parameters)
  - Find good values for the parameters s.t. the system behaves well

[Mar11] Nicolas Markey. "Robustness in Real-time Systems". In: *SIES*. IEEE Computer Society Press, June 2011, pp. 28–34

Parametric Timed Automaton (PTA)

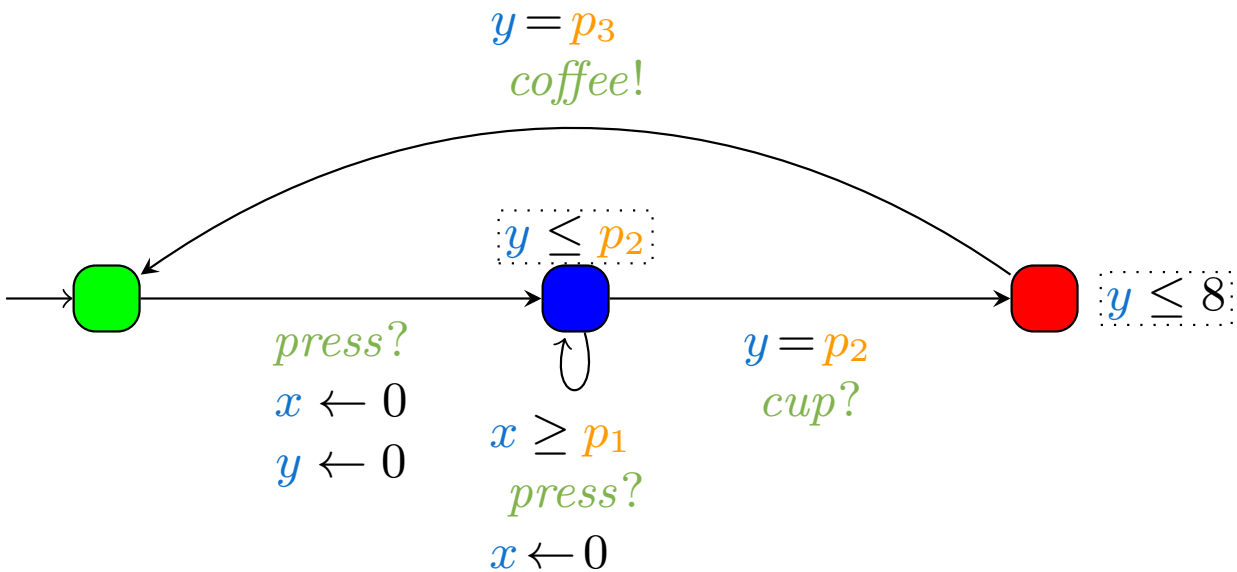
- Timed automaton (sets of locations, actions and clocks)



[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

Parametric Timed Automaton (PTA)

- Timed automaton (sets of locations, actions and clocks) augmented with a set P of parameters [AHV93]
  - Unknown constants compared to a clock in guards and invariants



[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

Parametric timed model checking in a nutshell

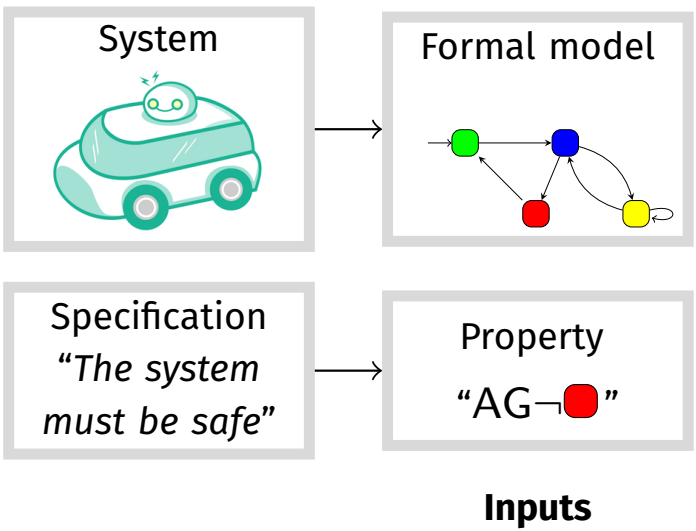
- Principle of parametric timed model checking

System

Specification
"The system must be safe"

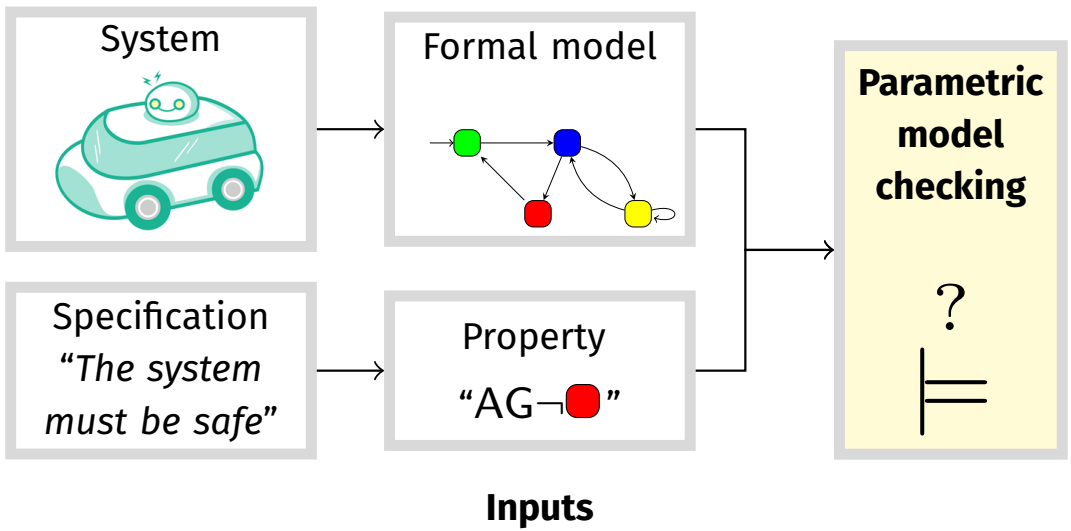
Parametric timed model checking in a nutshell

- Principle of parametric timed model checking



Parametric timed model checking in a nutshell

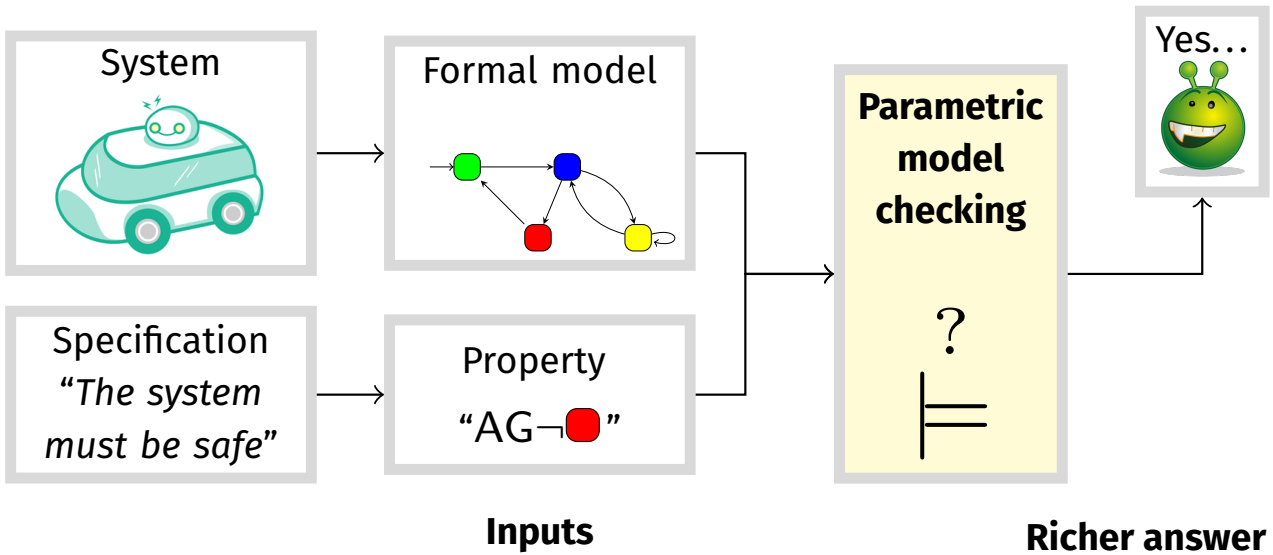
- Principle of parametric timed model checking



- Question: for which values of the design parameters does the model of the system satisfy the property?

Parametric timed model checking in a nutshell

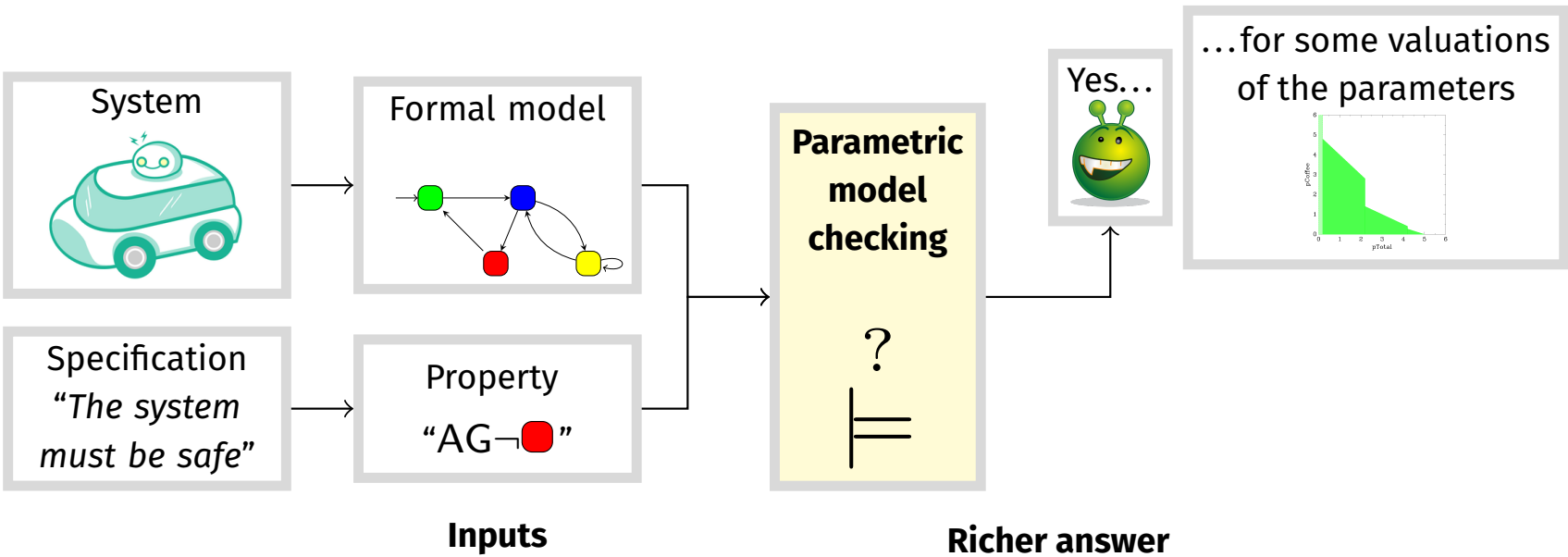
- Principle of parametric timed model checking



- Question: for which values of the design parameters does the model of the system satisfy the property?

Parametric timed model checking in a nutshell

- Principle of parametric timed model checking



- Question: for which values of the design parameters does the model of the system satisfy the property?

Parametric timed automata

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)

Parametric timed automata

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ uncertainty w.r.t. time

Parametric timed automata

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ uncertainty w.r.t. time
- ☹ mostly undecidable

Parametric timed automata

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ uncertainty w.r.t. time
- ☹ mostly undecidable
- ☹ fixed and *a priori* known number of agents

Outline

Need for uncertainty in the number of participants

The number of system agents may be unknown

- networks
- communication protocols
- leader election

Need for uncertainty in the number of participants

The number of system agents may be unknown

- networks
- communication protocols
- leader election

Natural reachability questions include

- can a location be reached by one agent for some number of agents?
- can a location be reached by all agents for some number of agents?

Need for uncertainty in the number of participants

The number of system agents may be unknown

- networks
- communication protocols
- leader election

Natural reachability questions include

- can a location be reached by one agent for some number of agents?
- can a location be reached by all agents for some number of agents?

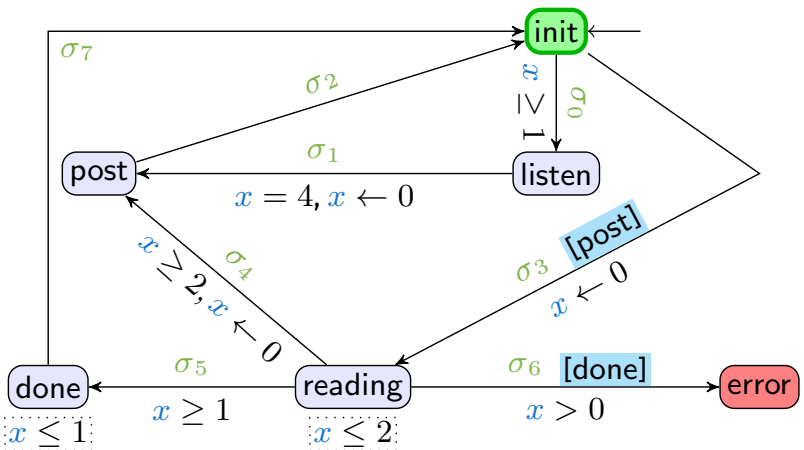
Considering networks of timed automata of arbitrary size:

☹ Many undecidable problems

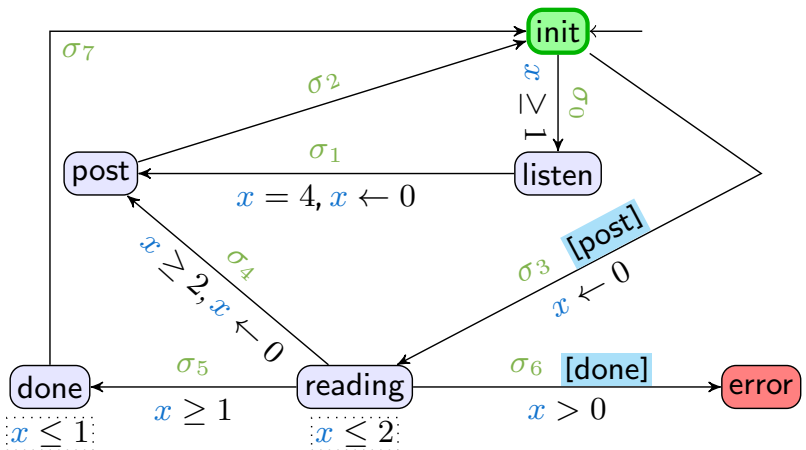
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 0$



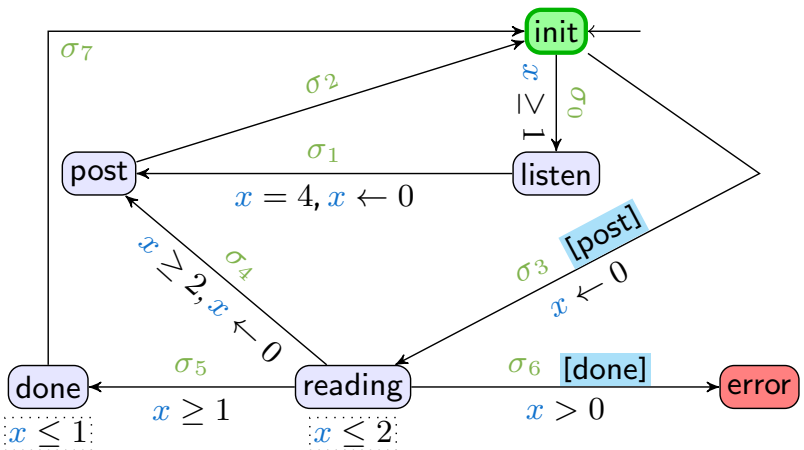
$x = 0$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

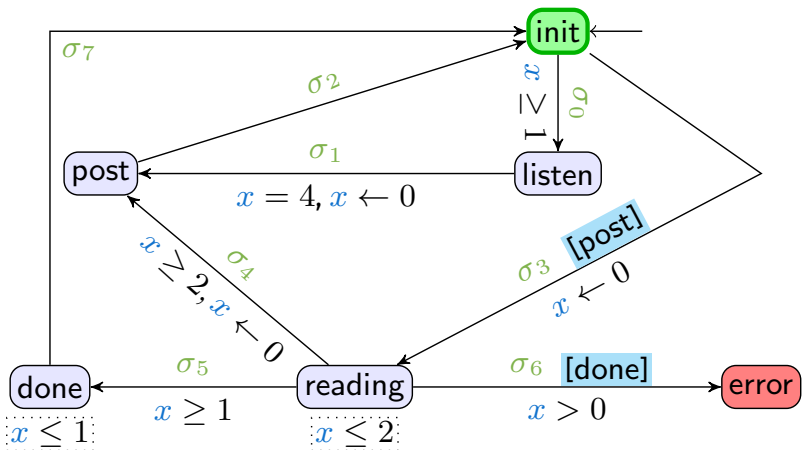
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 1.3$



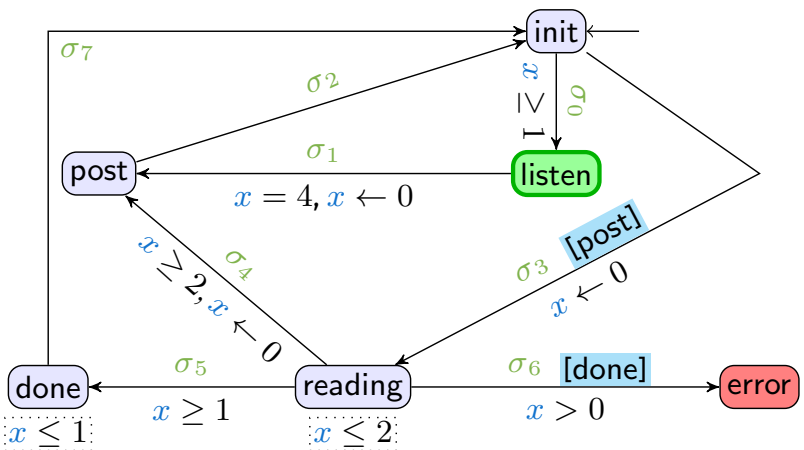
$x = 1.3$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

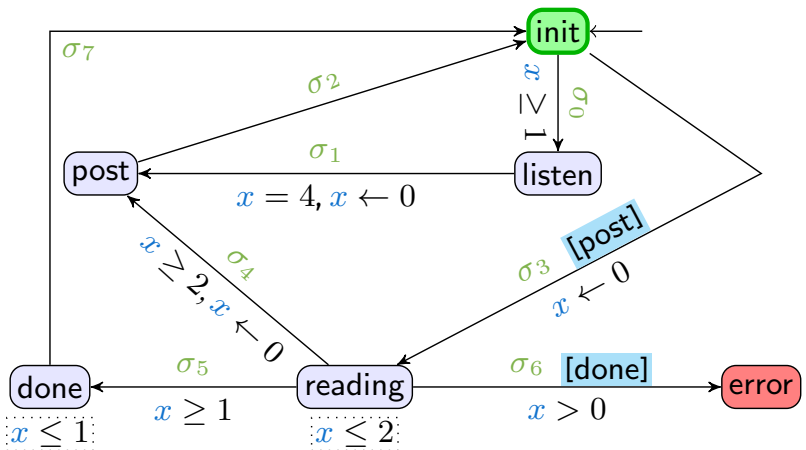
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 1.3$



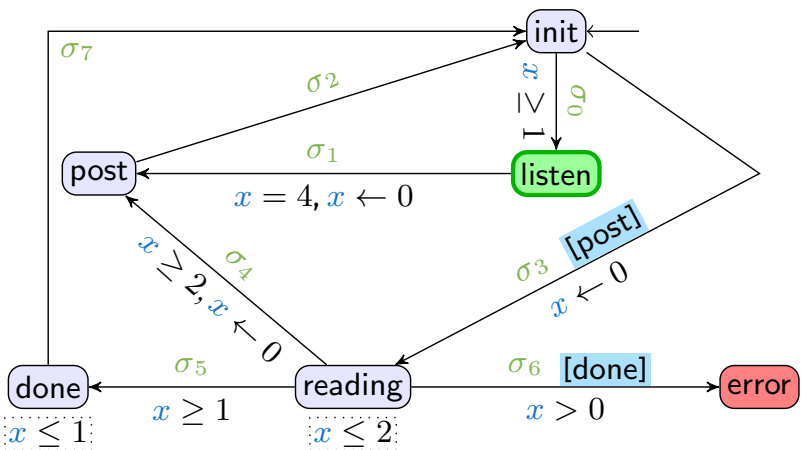
$x = 1.3$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

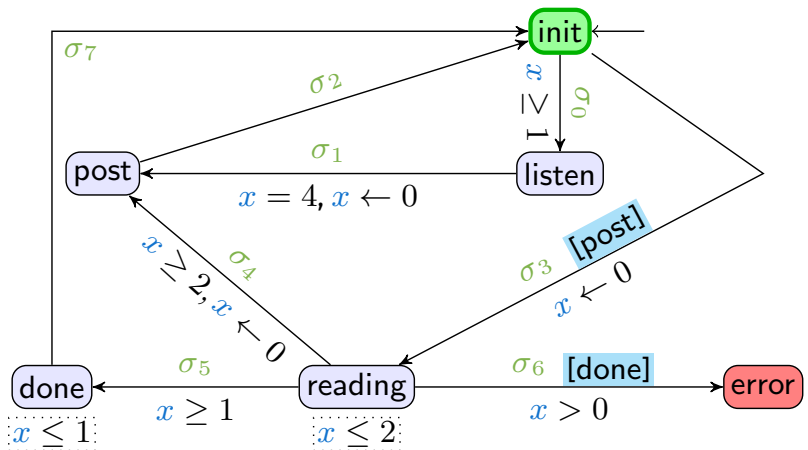
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 4$



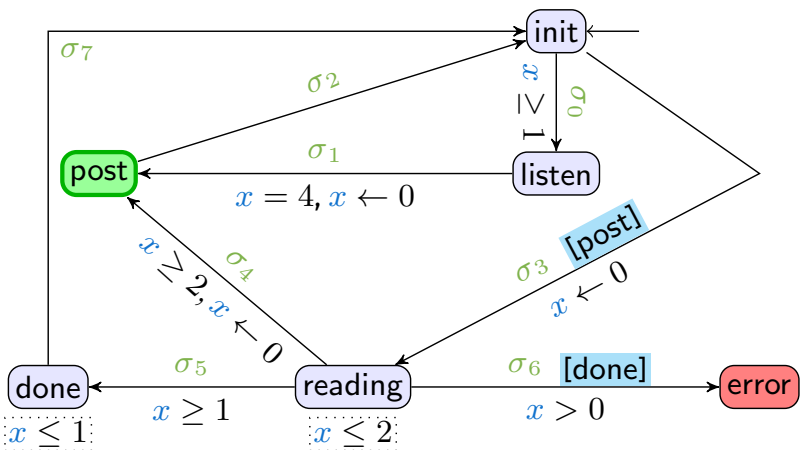
$x = 4$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

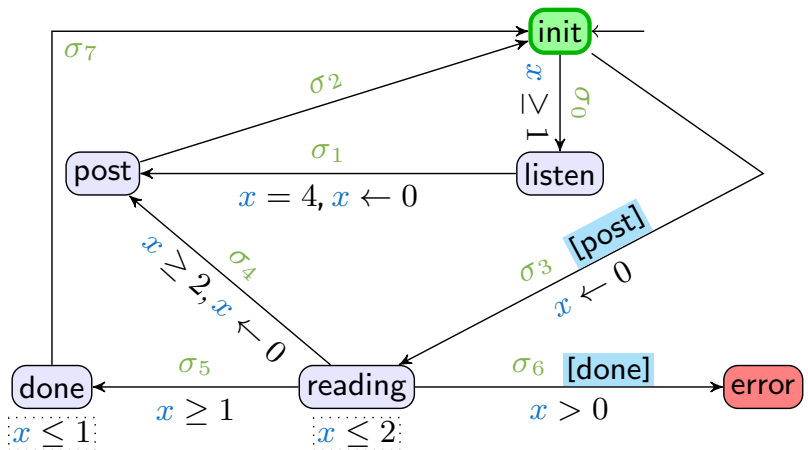
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 0$



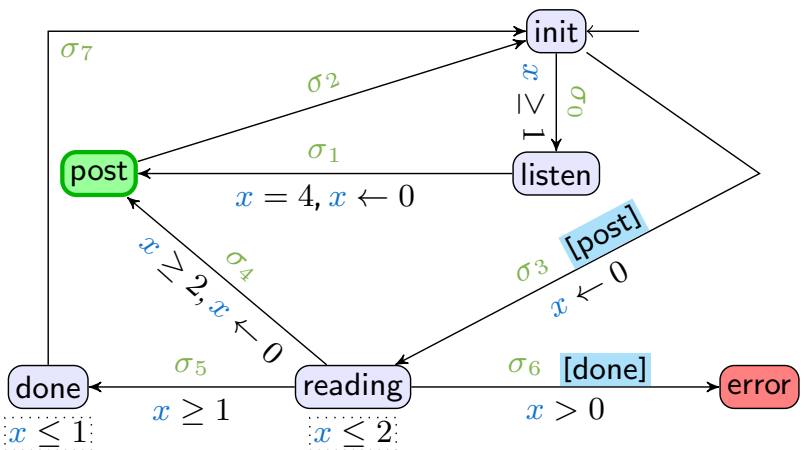
$x = 4$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

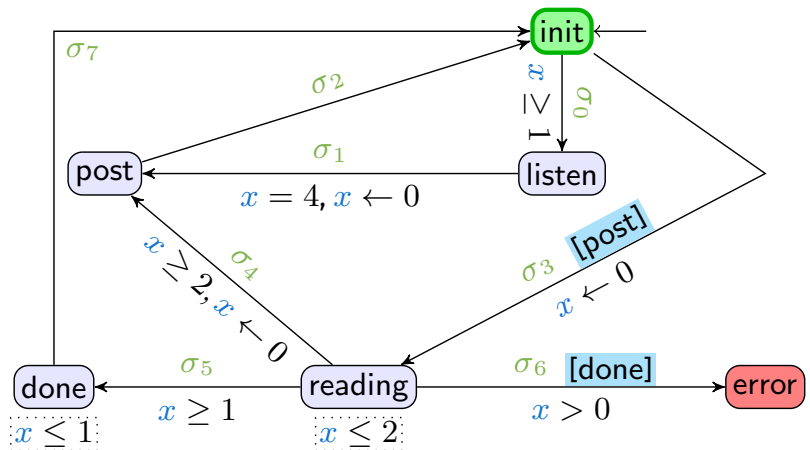
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 0.8$



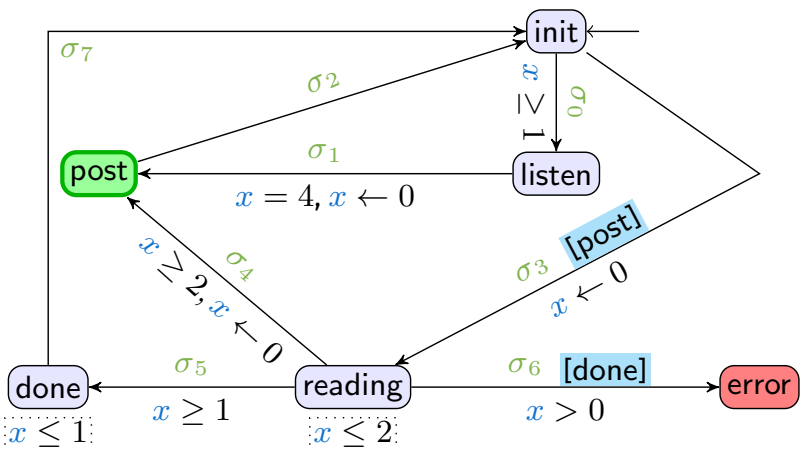
$x = 4.8$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

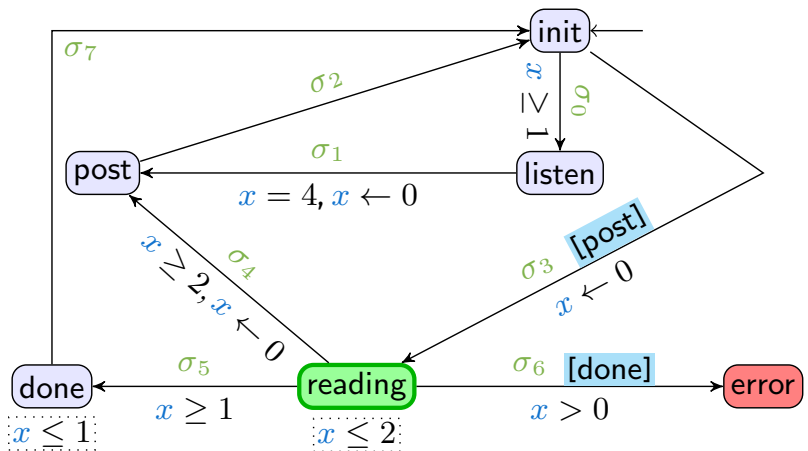
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 0.8$



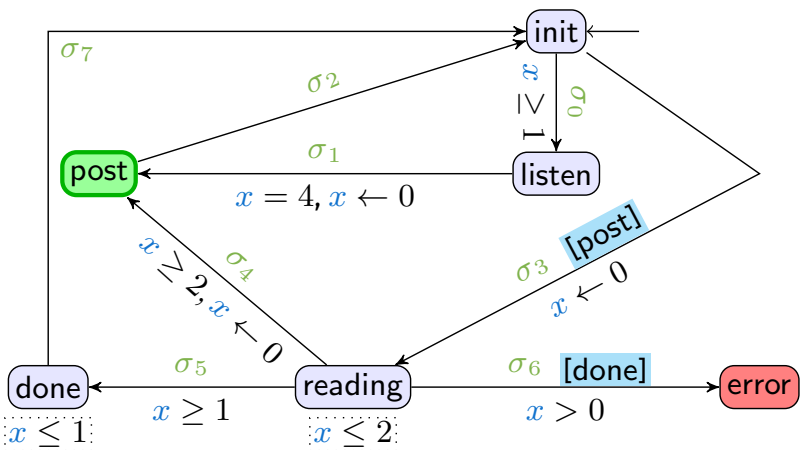
$x = 0$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

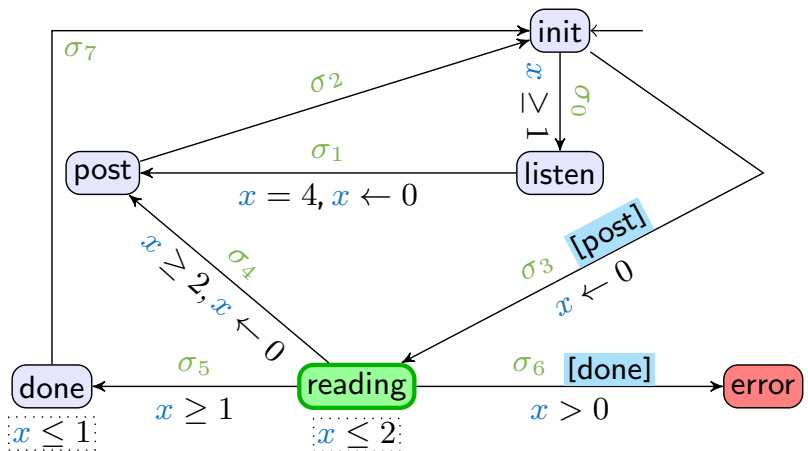
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 1.8$



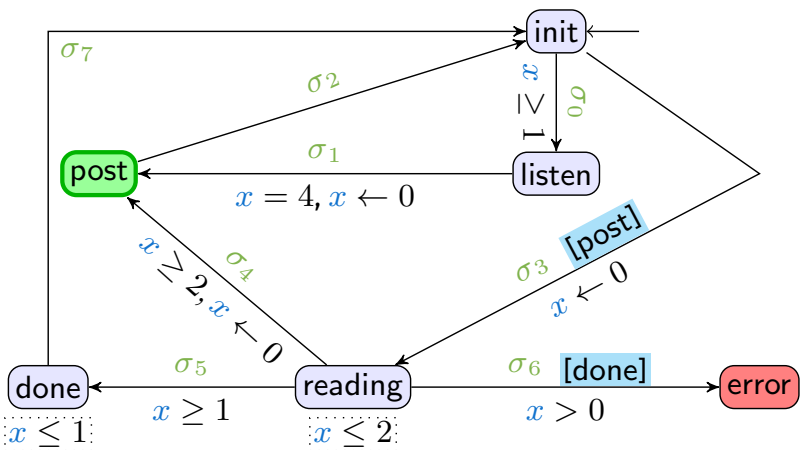
$x = 1$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

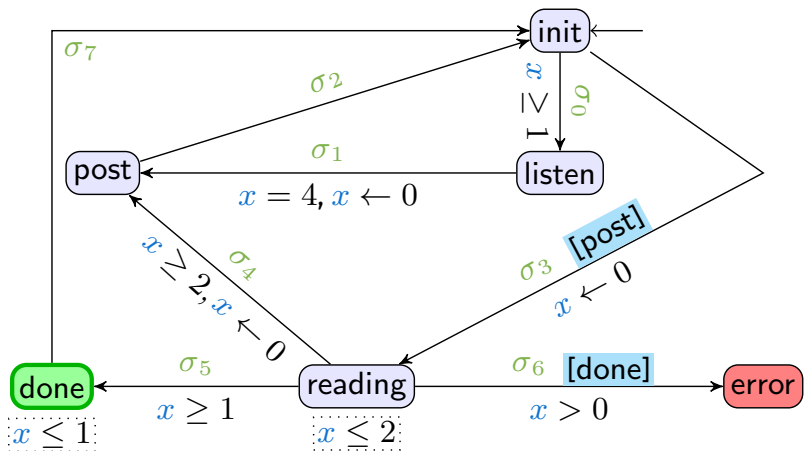
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



$x = 1.8$



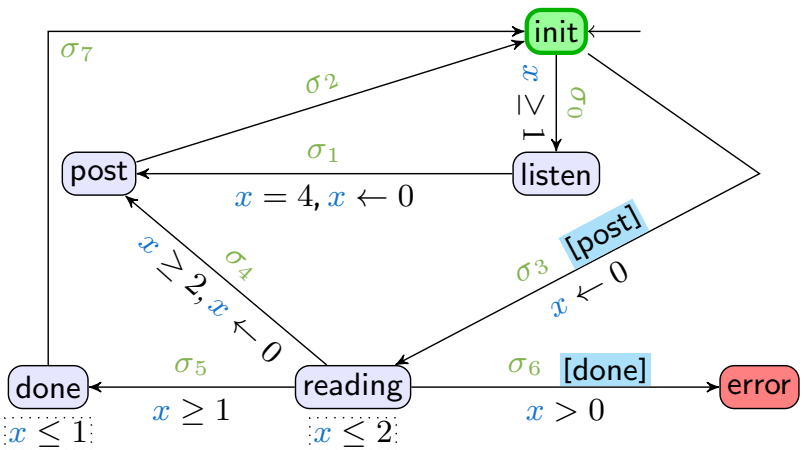
$x = 1$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

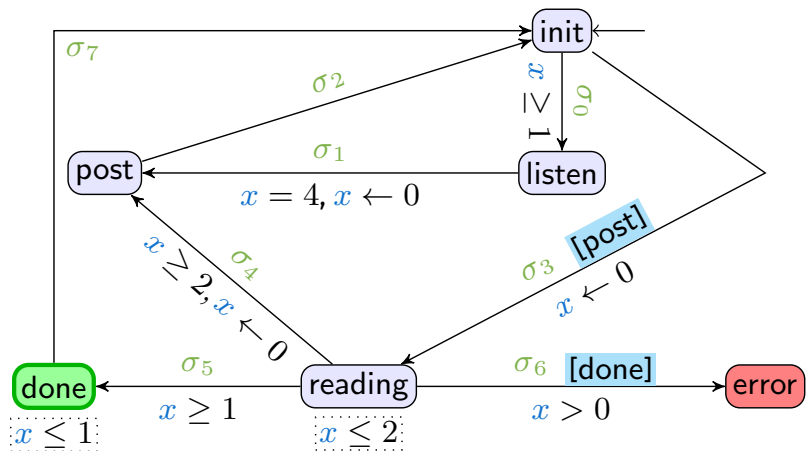
Disjunctive timed networks (DTNs)

Network of identical timed automata (“processes”) communicating via location guards [SS20]

- “a process can take a transition guarded by a location $[\ell]$ if at least one other process currently occupies ℓ”



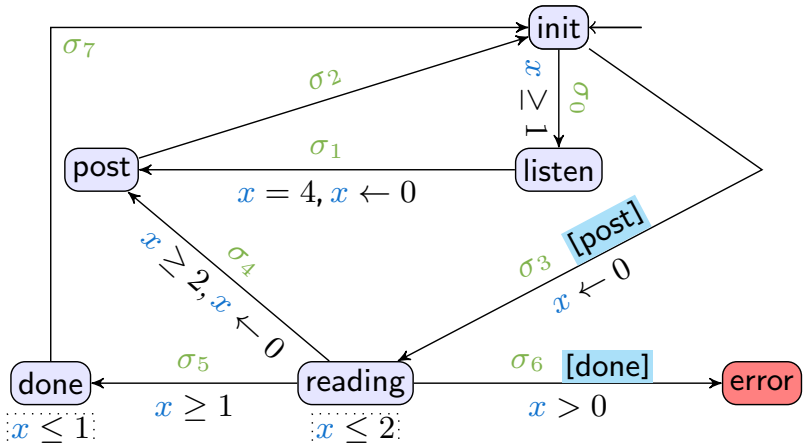
$x = 1.8$



$x = 1$

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

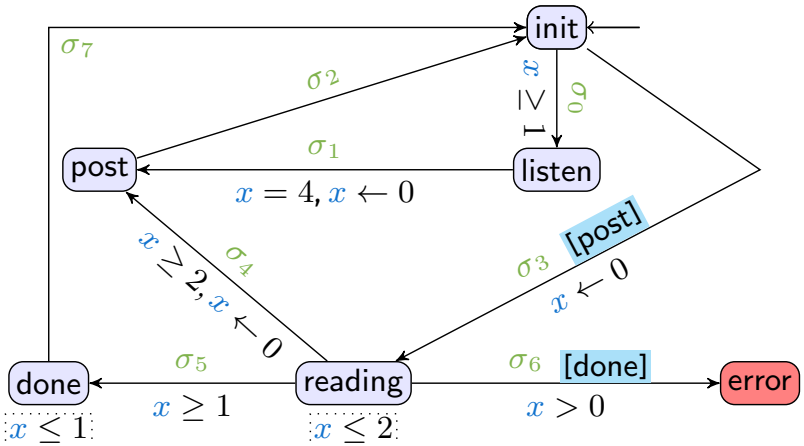
Disjunctive timed networks: problems



Example of questions

- Is location “error” reachable by one process for some number of processes?
  - 1 process:

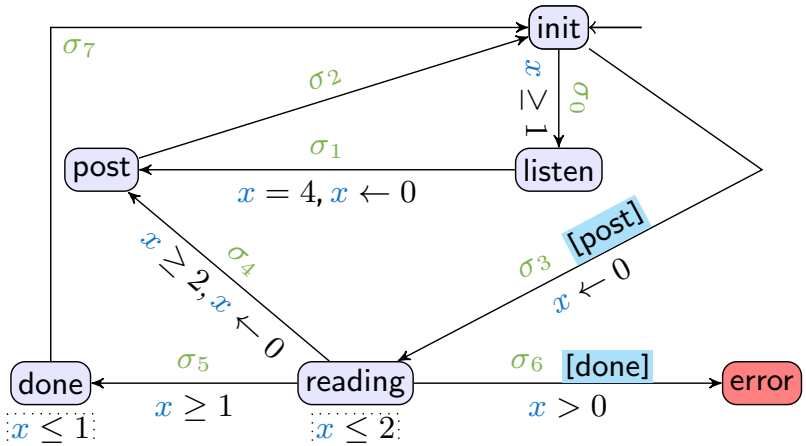
Disjunctive timed networks: problems



Example of questions

- Is location “error” reachable by one process for some number of processes?
  - 1 process: ×
  - 2 processes:

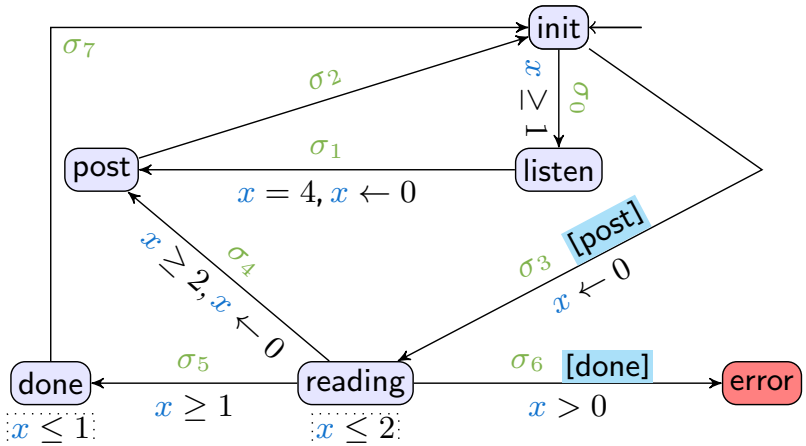
Disjunctive timed networks: problems



Example of questions

- Is location “error” reachable by one process for some number of processes?
  - 1 process: ×
  - 2 processes: ×
  - 3 processes:

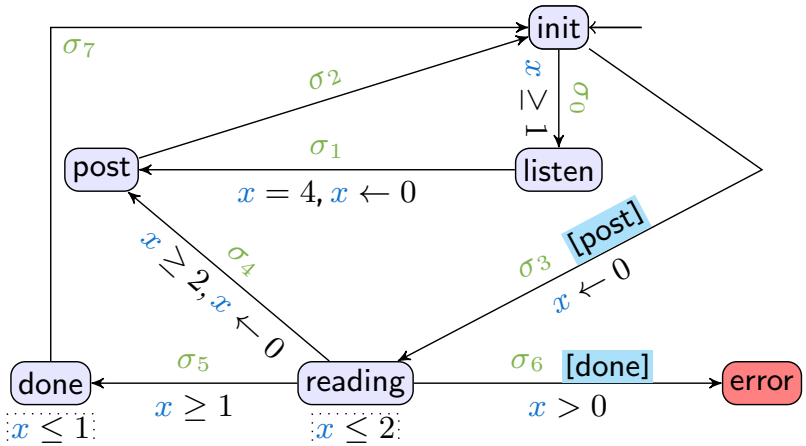
Disjunctive timed networks: problems



Example of questions

- Is location “error” reachable by one process for some number of processes?
  - 1 process: ×
  - 2 processes: ×
  - 3 processes: √
  - ≥ 3 processes:

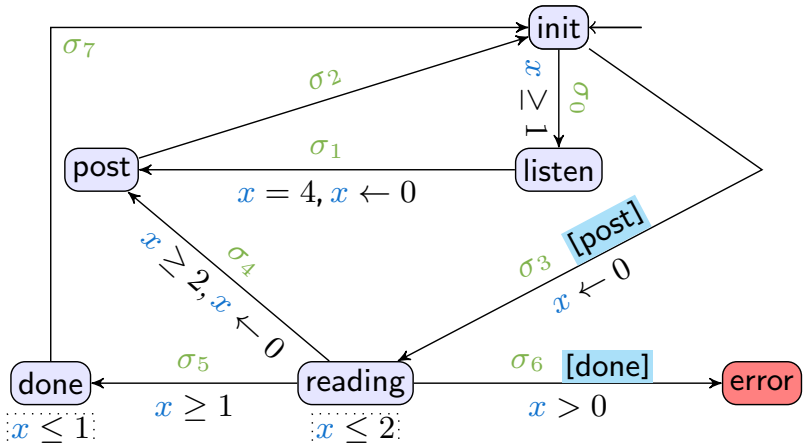
Disjunctive timed networks: problems



Example of questions

- Is location “error” reachable by one process for some number of processes?
  - 1 process: ×
  - 2 processes: ×
  - 3 processes: √
  - ≥ 3 processes: √
- Is location “error” reachable by all processes for some number of processes?

Disjunctive timed networks: problems



Example of questions

- Is location “error” reachable by one process for some number of processes?
  - 1 process: ×
  - 2 processes: ×
  - 3 processes: √
  - ≥ 3 processes: √

- Is location “error” reachable by all processes for some number of processes?
  - ×

Disjunctive timed networks: decidability

Invariants	Local properties	Global properties
Without	$\surd$ [SS20][And+24]	$(?)$
With	$\surd$ [And+25]	$(?)$

Informal definitions:

Local properties “reachability for one process”

Global properties “reachability for all processes”

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269

[And+24] Étienne André, Paul Eichler, Swen Jacobs, and Shyam Lal Karra. “Parameterized Verification of Disjunctive Timed Networks”. In: *VMCAI*. vol. 14499. Lecture Notes in Computer Science. Springer, 2024, pp. 124–146

[And+25] Étienne André, Swen Jacobs, Shyam Lal Karra, and Ocan Sankur. “Parameterized Verification of Timed Networks with Clock Invariants”. In: *FSTTCS*. vol. 360. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Dec. 2025, 8:1–8:19

Disjunctive timed networks: advantages and drawbacks

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ some decidable problems

Disjunctive timed networks: advantages and drawbacks

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ some decidable problems
- ☺ arbitrary number of agents

Disjunctive timed networks: advantages and drawbacks

- ☺ multiple agents
- ☺ concurrency
- ☺ synchronisation
- ☺ real-time constraints (clocks)
- ☺ some decidable problems
- ☺ arbitrary number of agents
- ☹ requires infinite precision w.r.t. time

Outline

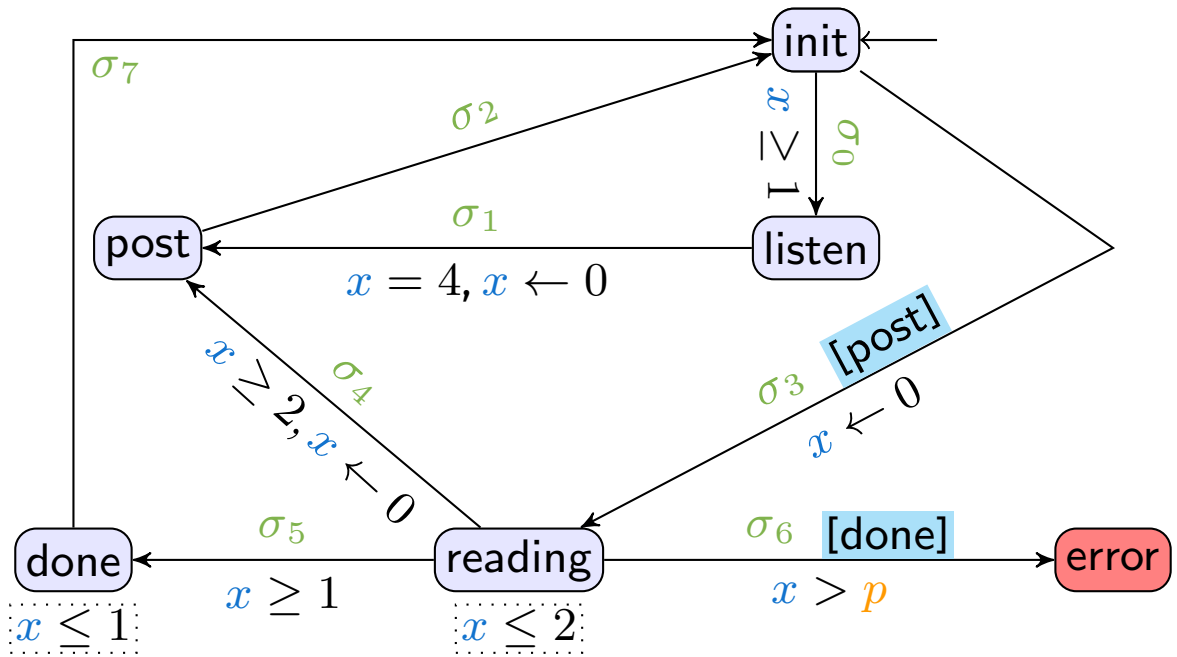
Objective: combine both types of uncertainty

Consider an unbounded number of agents under uncertainty over time

Objective: combine both types of uncertainty

Consider an unbounded number of agents under uncertainty over time

- We introduce parametric disjunctive timed networks (PDTNs)



Challenge

Many existing undecidability results when considering only one of our two types of uncertainty in isolation

- ☹ single parametric timed automaton with ≥ 3 clocks
- ☹ single parametric timed automaton with 1 clock compared to 1 parameter (and 3 other clocks)
- ☹ networks of arbitrary numbers of non-parametric timed automata with other synchronisation mechanisms than location guards

Challenge

Many existing undecidability results when considering only one of our two types of uncertainty in isolation

- ☹ single parametric timed automaton with ≥ 3 clocks
- ☹ single parametric timed automaton with 1 clock compared to 1 parameter (and 3 other clocks)
- ☹ networks of arbitrary numbers of non-parametric timed automata with other synchronisation mechanisms than location guards

Challenge: exhibit decidability results for subclasses of PDTNs

Main existential problem: "does there exist a number of processes and a valuation of the timing parameters such that one process can reach a given location?"

Challenge

Many existing undecidability results when considering only one of our two types of uncertainty in isolation

- ☹ single parametric timed automaton with ≥ 3 clocks
- ☹ single parametric timed automaton with 1 clock compared to 1 parameter (and 3 other clocks)
- ☹ networks of arbitrary numbers of non-parametric timed automata with other synchronisation mechanisms than location guards

Challenge: exhibit decidability results for subclasses of PDTNs

Main existential problem: "does there exist a number of processes and a valuation of the timing parameters such that one process can reach a given location?"

Useful notably for safety

Outline

Our paper in one slide

Clocks	Integer-valued parameters	Invariants	Local properties	Global properties
1	arbitrary	**Without**	√	
		With		
arbitrary	1-param. fully-parametric	**Without**		
		With		
arbitrary	arbitrary, L/U	**Without**		
		With		

☺ Main decidability result: formula in Presburger arithmetic enhanced with the divisibility operand, written using affine parametric semi-linear sets

Our paper in one slide

Clocks	Integer-valued parameters	Invariants	Local properties	Global properties
1	arbitrary	**Without**	$\surd$	
		With	$\times$	
arbitrary	1-param. fully-parametric	**Without**		
		With		
arbitrary	arbitrary, L/U	**Without**		
		With		

- ☺ Main decidability result: formula in Presburger arithmetic enhanced with the divisibility operand, written using affine parametric semi-linear sets
- ☹ Main undecidability results: 2-counter machine simulated by forcing synchronisation between several 1-clock PTAs over 1 parameter using
  - either invariants to avoid processes “left behind”

Our paper in one slide

Clocks	Integer-valued parameters	Invariants	Local properties	Global properties
1	arbitrary	**Without**	$\surd$	$\times$
		With	$\times$	$\times$
arbitrary	1-param. fully-parametric	**Without**		
		With		
arbitrary	arbitrary, L/U	**Without**		
		With		

- ☺ Main decidability result: formula in Presburger arithmetic enhanced with the divisibility operand, written using affine parametric semi-linear sets
- ☹ Main undecidability results: 2-counter machine simulated by forcing synchronisation between several 1-clock PTAs over 1 parameter using
  - either invariants to avoid processes “left behind”
  - or global properties to ensure all processes reach the halting location

Our paper in one slide

Clocks	Integer-valued parameters	Invariants	Local properties	Global properties
1	arbitrary	**Without**	$\surd$	$\times$ ☻
		With	$\times$ ☻	$\times$
arbitrary	1-param. fully-parametric	**Without**		
		With		
arbitrary	arbitrary, L/U	**Without**		
		With		

- ☺ Main decidability result: formula in Presburger arithmetic enhanced with the divisibility operand, written using affine parametric semi-linear sets
- ☹ Main undecidability results: 2-counter machine simulated by forcing synchronisation between several 1-clock PTAs over 1 parameter using
  - either invariants to avoid processes “left behind”
  - or global properties to ensure all processes reach the halting location
  - Our weak communication primitive increases expressiveness significantly

Our paper in one slide

Clocks	Integer-valued parameters	Invariants	Local properties	Global properties
1	arbitrary	**Without**	$\surd$	$\times$ ☻
		With	$\times$ ☻	$\times$
arbitrary	1-param. fully-parametric	**Without**	$\surd$	
		With	$\surd$	
arbitrary	arbitrary, L/U	**Without**		
		With		

- ☺ Main decidability result: formula in Presburger arithmetic enhanced with the divisibility operand, written using affine parametric semi-linear sets
- ☹ Main undecidability results: 2-counter machine simulated by forcing synchronisation between several 1-clock PTAs over 1 parameter using
  - either invariants to avoid processes "left behind"
  - or global properties to ensure all processes reach the halting location
  - Our weak communication primitive increases expressiveness significantly
- ☺ Also decidable: single parameter, no constant (apart from 0) in the PTAs

Our paper in one slide

Clocks	Integer-valued parameters	Invariants	Local properties	Global properties
1	arbitrary	**Without**	√	× ☻
		With	× ☻	×
arbitrary	1-param. fully-parametric	**Without**	√	
		With	√	
arbitrary	arbitrary, L/U	**Without**	√	
		With	√	

- ☺ Main decidability result: formula in Presburger arithmetic enhanced with the divisibility operand, written using affine parametric semi-linear sets
- ☹ Main undecidability results: 2-counter machine simulated by forcing synchronisation between several 1-clock PTAs over 1 parameter using
  - either invariants to avoid processes "left behind"
  - or global properties to ensure all processes reach the halting location
  - Our weak communication primitive increases expressiveness significantly
- ☺ Also decidable: single parameter, no constant (apart from 0) in the PTAs
- ☺ Also decidable: parameters partitioned into lower-bound and upper-bound parameters
  - monotonicity: boils down to non-parametric DTNs

Our paper in one slide

Clocks	Integer-valued parameters	Invariants	Local properties	Global properties
1	arbitrary	**Without**	$\sqrt{}$	$\times$ ☻
		With	$\times$ ☻	$\times$
arbitrary	1-param. fully-parametric	**Without**	$\sqrt{}$	?
		With	$\sqrt{}$	?
arbitrary	arbitrary, L/U	**Without**	$\sqrt{}$	?
		With	$\sqrt{}$	?

- ☺ Main decidability result: formula in Presburger arithmetic enhanced with the divisibility operand, written using affine parametric semi-linear sets
- ☹ Main undecidability results: 2-counter machine simulated by forcing synchronisation between several 1-clock PTAs over 1 parameter using
  - either invariants to avoid processes "left behind"
  - or global properties to ensure all processes reach the halting location
  - Our weak communication primitive increases expressiveness significantly
- ☺ Also decidable: single parameter, no constant (apart from 0) in the PTAs
- ☺ Also decidable: parameters partitioned into lower-bound and upper-bound parameters
  - monotonicity: boils down to non-parametric DTNs

Outline

General perspectives

Open decidability problems

- Global properties for our decidable cases
  - Do all processes reach a given location for at least one valuation of the parameters and at least one number of processes?

General perspectives

Open decidability problems

- Global properties for our decidable cases
  - Do all processes reach a given location for at least one valuation of the parameters and at least one number of processes?
- Universality problems
  - Does a property hold for all valuations of the parameters for at least one number of processes?
  - Does a property hold for at least one valuation of the parameters for all numbers of processes?

General perspectives

Open decidability problems

- Global properties for our decidable cases
  - Do all processes reach a given location for at least one valuation of the parameters and at least one number of processes?
- Universality problems
  - Does a property hold for all valuations of the parameters for at least one number of processes?
  - Does a property hold for at least one valuation of the parameters for all numbers of processes?
- (Bounded) rational-valued timing parameters

General perspectives

Open decidability problems

- Global properties for our decidable cases
  - Do all processes reach a given location for at least one valuation of the parameters and at least one number of processes?
- Universality problems
  - Does a property hold for all valuations of the parameters for at least one number of processes?
  - Does a property hold for at least one valuation of the parameters for all numbers of processes?
- (Bounded) rational-valued timing parameters

Synthesis

- Synthesize all timing parameters valuations such that a property holds
  - not easy! likely undoable for our decidable L/U subclass

General perspectives

Open decidability problems

- Global properties for our decidable cases
  - Do all processes reach a given location for at least one valuation of the parameters and at least one number of processes?
- Universality problems
  - Does a property hold for all valuations of the parameters for at least one number of processes?
  - Does a property hold for at least one valuation of the parameters for all numbers of processes?
- (Bounded) rational-valued timing parameters

Synthesis

- Synthesize all timing parameters valuations such that a property holds
  - not easy! likely undoable for our decidable L/U subclass
- Implementation

Bibliography

References I

[AAC12] Parosh Aziz Abdulla, Mohamed Faouzi Atig, and Jonathan Cederberg. "Timed Lossy Channel Systems". In: *FSTTCS* (Dec. 15–17, 2012). Ed. by Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan. Vol. 18. LIPIcs. Hyderabad, India: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012, pp. 374–386. DOI: 10.4230/LIPICS.FSTTCS.2012.374.

[Abd+16] Parosh Aziz Abdulla, Giorgio Delzanno, Othmane Rezine, Arnaud Sangnier, and Riccardo Traverso. "Parameterized verification of time-sensitive models of ad hoc network protocols". In: *Theoretical Computer Science* 612 (2016), pp. 1–22. DOI: 10.1016/j.tcs.2015.07.048.

[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. DOI: 10.1016/0304-3975(94)90010-8.

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC* (May 16–18, 1993). Ed. by S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal. San Diego, California, USA: ACM, 1993, pp. 592–601. DOI: 10.1145/167088.167242.

[AJ03] Parosh Aziz Abdulla and Bengt Jonsson. "Model checking of systems with many identical timed processes". In: *Theoretical Computer Science* 290.1 (2003), pp. 241–264. DOI: 10.1016/S0304-3975(01)00330-9.

[AK86] Krzysztof R. Apt and Dexter Kozen. "Limits for Automatic Verification of Finite-State Concurrent Systems". In: *Information Processing Letters* 22.6 (1986), pp. 307–309. DOI: 10.1016/0020-0190(86)90071-2.

References II

[Ami+15] Benjamin Aminof, Sasha Rubin, Florian Zuleger, and Francesco Spegni. "Liveness of Parameterized Timed Networks". In: *ICALP, Part II* (July 6–10, 2015). Ed. by Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann. Vol. 9135. Lecture Notes in Computer Science. Kyoto, Japan: Springer, 2015, pp. 375–387. DOI: 10.1007/978-3-662-47666-6_30.

[And+19] Étienne André, Benoît Delahaye, Paulin Fournier, and Didier Lime. "Parametric Timed Broadcast Protocols". In: *VMCAI* (Jan. 13–15, 2019). Ed. by Constantin Enea and Ruzica Piskac. Vol. 11388. Lecture Notes in Computer Science. Lisbon, Portugal: Springer, 2019, pp. 491–512. DOI: 10.1007/978-3-030-11245-5_23.

[And+24] Étienne André, Paul Eichler, Swen Jacobs, and Shyam Lal Karra. "Parameterized Verification of Disjunctive Timed Networks". In: *VMCAI* (Jan. 15–16, 2024). Ed. by Rayna Dimitrova and Ori Lahav. Vol. 14499. Lecture Notes in Computer Science. London, England: Springer, 2024, pp. 124–146. DOI: 10.1007/978-3-031-50524-9_6.

[And+25] Étienne André, Swen Jacobs, Shyam Lal Karra, and Ocan Sankur. "Parameterized Verification of Timed Networks with Clock Invariants". In: *FSTTCS* (Dec. 17–19, 2025). Ed. by C Aiswarya, Ruta Mehta, and Subhajit Roy. Vol. 360. LIPIcs. Goa, India: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Dec. 2025, 8:1–8:19. DOI: 10.4230/LIPIcs.FSTTCS.2025.8.

[BF13] Nathalie Bertrand and Paulin Fournier. "Parameterized Verification of Many Identical Probabilistic Timed Processes". In: *FSTTCS* (Dec. 12–14, 2013). Ed. by Anil Seth and Nisheeth K. Vishnoi. Vol. 24. LIPIcs. Guwahati, India: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013, pp. 501–513. DOI: 10.4230/LIPIcs.FSTTCS.2013.501.

References III

[EK00] E. Allen Emerson and Vineet Kahlon. “Reducing Model Checking of the Many to the Few”. In: *CADE* (June 17–20, 2000). Ed. by David A. McAllester. Vol. 1831. Lecture Notes in Computer Science. Pittsburgh, PA, USA: Springer, 2000, pp. 236–254. DOI: 10.1007/10721959_19.

[Mar11] Nicolas Markey. “Robustness in Real-time Systems”. In: *SIES*. Ed. by Iain Bate and Roberto Passerone. Västerås, Sweden: IEEE Computer Society Press, June 2011, pp. 28–34. DOI: 10.1109/SIES.2011.5953652.

[SS20] Luca Spalazzi and Francesco Spegni. “Parameterized model checking of networks of timed automata with Boolean guards”. In: *Theoretical Computer Science* 813 (2020), pp. 248–269. DOI: 10.1016/j.tcs.2019.12.026.

[Suz88] Ichiro Suzuki. “Proving Properties of a Ring of Finite-State Machines”. In: *Information Processing Letters* 28.4 (1988), pp. 213–214. DOI: 10.1016/0020-0190(88)90211-6.

Additional slides

Parameterized verification

Undecidable in general [AK86]

- Even in very restricted settings (for safety properties of finite-state processes with rather weak communication primitives, such as token-passing or transition guards) [Suz88][EK00]

[AK86] Krzysztof R. Apt and Dexter Kozen. "Limits for Automatic Verification of Finite-State Concurrent Systems". In: *Information Processing Letters* 22.6 (1986), pp. 307–309

[Suz88] Ichiro Suzuki. "Proving Properties of a Ring of Finite-State Machines". In: *Information Processing Letters* 28.4 (1988), pp. 213–214

[EK00] E. Allen Emerson and Vineet Kahlon. "Reducing Model Checking of the Many to the Few". In: *CADE*. vol. 1831. Lecture Notes in Computer Science. Springer, 2000, pp. 236–254

Parameterized timed verification

Several lines of research studying networks with a parametric number of timed components, including

- [AJ03]
- [AAC12]
- [BF13]
- [Ami+15]
- [Abd+16]
- [And+19]

[AJ03] Parosh Aziz Abdulla and Bengt Jonsson. "Model checking of systems with many identical timed processes". In: *Theoretical Computer Science* 290.1 (2003), pp. 241–264

[AAC12] Parosh Aziz Abdulla, Mohamed Faouzi Atig, and Jonathan Cederberg. "Timed Lossy Channel Systems". In: *FSTTCS*. vol. 18. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012, pp. 374–386

[BF13] Nathalie Bertrand and Paulin Fournier. "Parameterized Verification of Many Identical Probabilistic Timed Processes". In: *FSTTCS*. vol. 24. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013, pp. 501–513

[Ami+15] Benjamin Aminof, Sasha Rubin, Florian Zuleger, and Francesco Spegni. "Liveness of Parameterized Timed Networks". In: *ICALP, Part II*. vol. 9135. Lecture Notes in Computer Science. Springer, 2015, pp. 375–387

[Abd+16] Parosh Aziz Abdulla, Giorgio Delzanno, Othmane Rezine, Arnaud Sangnier, and Riccardo Traverso. "Parameterized verification of time-sensitive models of ad hoc network protocols". In: *Theoretical Computer Science* 612 (2016), pp. 1–22

[And+19] Étienne André, Benoît Delahaye, Paulin Fournier, and Didier Lime. "Parametric Timed Broadcast Protocols". In: *VMCAI*. vol. 11388. Lecture Notes in Computer Science. Springer, 2019, pp. 491–512

Licensing

Source of the graphics used I

Title : Autonomous robot vehicle or ADV typically used for food or grocery delivery
Author : Rlistmedia
Source : `https://commons.wikimedia.org/wiki/File:Autonomous_delivery_robot_vehicles_ADV.png`
License : Creative Commons cc-by

Title : Smiley green alien big eyes (aaah)
Author : LadyofHats
Source : `https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg`
License : Public domain

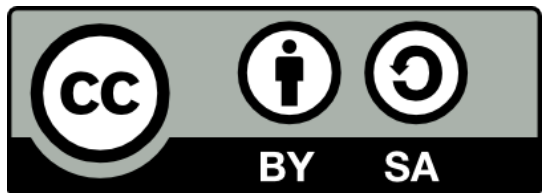
Title : Smiley green alien big eyes (cry)
Author : LadyofHats
Source : `https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg`
License : Public domain

License of this document

This presentation can be published, reused and modified under the terms of the license Creative Commons **Attribution-ShareAlike 4.0 Unported (CC BY-SA 4.0)**

(LaTeX source available upon request)

Authors: **Étienne André**



creativecommons.org/licenses/by-sa/4.0/