CAV 2021

July 2021

IMITATOR 3

Synthesis of timing parameters beyond decidability

Étienne André

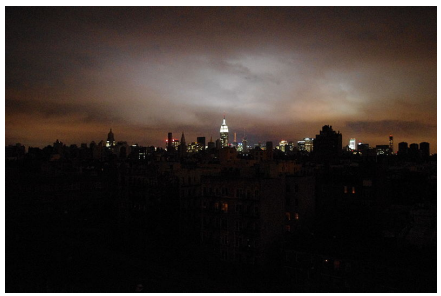
Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

Supported by the ANR-NRF research program ProMiS (ANR-19-CE25-0015)

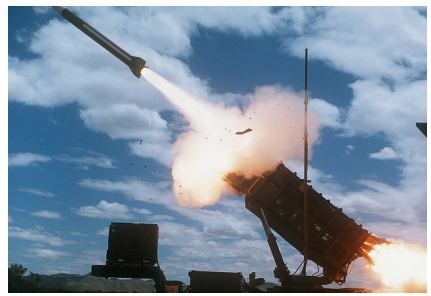
Context: Verifying critical real-time systems

real-time systems:

- Systems for which not only the functional correctness but also the timely answer is important



Northeast blackout
(USA, 2003)

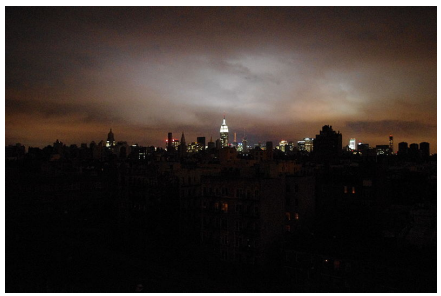


MIM-104 Patriot Missile Failure
(Iraq, 1991)

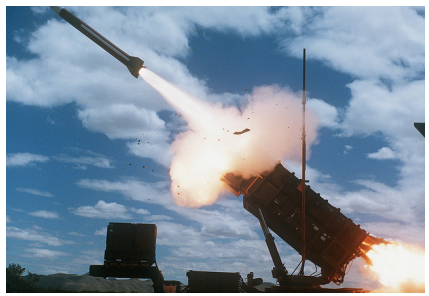
Sleipner A offshore platform
(Norway, 1991)

Context: Verifying critical real-time systems

- Critical real-time systems:
  - Systems for which not only the functional correctness but also the timely answer is important
  - Failures (in correctness or timing) may result in dramatic consequences



Northeast blackout
(USA, 2003)



MIM-104 Patriot Missile Failure
(Iraq, 1991)

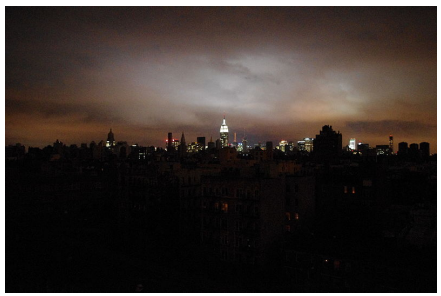
Sleipner A offshore platform
(Norway, 1991)

- Verification is needed to ensure the absence of bugs

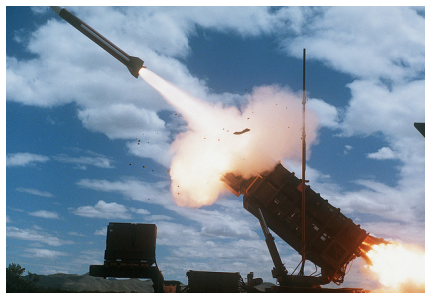
- Verification techniques
  - Testing
  - Abstract interpretation
  - Theorem proving
  - Model checking

Context: Verifying critical real-time systems

- Critical real-time systems:
  - Systems for which not only the functional correctness but also the timely answer is important
  - Failures (in correctness or timing) may result in dramatic consequences



Northeast blackout
(USA, 2003)



MIM-104 Patriot Missile Failure
(Iraq, 1991)

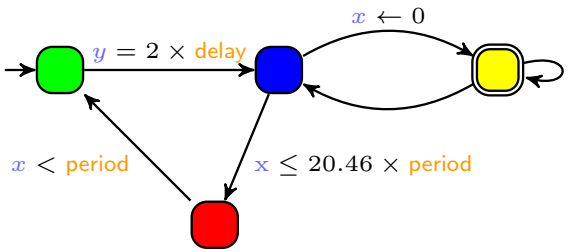
Sleipner A offshore platform
(Norway, 1991)

- Verification is needed to ensure the absence of bugs

- Verification techniques
  - Testing
  - Abstract interpretation
  - Theorem proving
  - Model checking

Model checking timed concurrent systems

- Principle of model checking



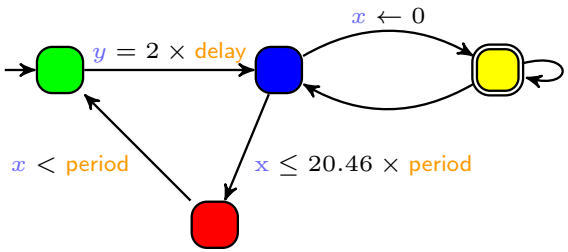
A model of the system

is unreachable

A property to be verified

Model checking timed concurrent systems

- Principle of model checking



A model of the system

$\models ?$

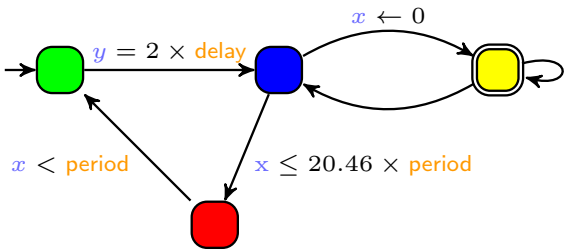
is unreachable

A property to be verified

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

Model checking timed concurrent systems

- Principle of model checking



A model of the system

$\models$?

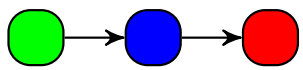
is unreachable

A property to be verified

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

Yes

No



Counterexample

Beyond timed model checking: parameter synthesis

- Verification for one set of constants does not usually guarantee the correctness for other values

- Challenges
  - Numerous verifications: is the system correct for any value within $[40; 60]$?
  - Optimization: until what value can we increase a given constant while preserving correctness?
  - Robustness [BMS13]: What happens if 50 is implemented with 49.99?
  - System incompletely specified: Can I verify my system even if I don't know the period value with full certainty?

[BMS13] Patricia Bouyer, Nicolas Markey, and Ocan Sankur. "Robustness in timed automata". In: *RP*. vol. 8169. LNCS. Invited paper. Springer, Sept. 2013, pp. 1–18

Beyond timed model checking: parameter synthesis

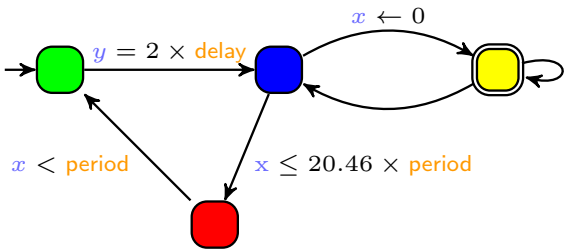
- Verification for one set of constants does not usually guarantee the correctness for other values

- Challenges
  - Numerous verifications: is the system correct for any value within $[40; 60]$?
  - Optimization: until what value can we increase a given constant while preserving correctness?
  - Robustness [BMS13]: What happens if 50 is implemented with 49.99?
  - System incompletely specified: Can I verify my system even if I don't know the period value with full certainty?

- Parameter synthesis
  - Consider that timing constants are unknown constants (parameters)

[BMS13] Patricia Bouyer, Nicolas Markey, and Ocan Sankur. "Robustness in timed automata". In: *RP*. vol. 8169. LNCS. Invited paper. Springer, Sept. 2013, pp. 1–18

timed model checking



A model of the system

$\models$?

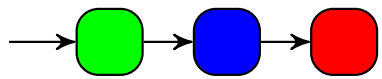
is unreachable

A property to be verified

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

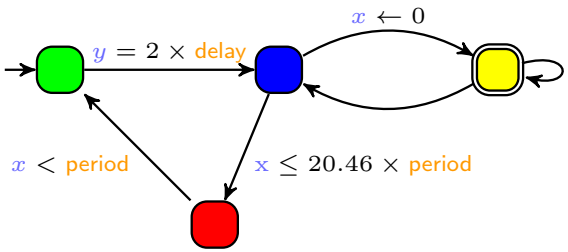
Yes

No



Counterexample

Parametric timed model checking



A model of the system

$\models$?

is unreachable

A property to be verified

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

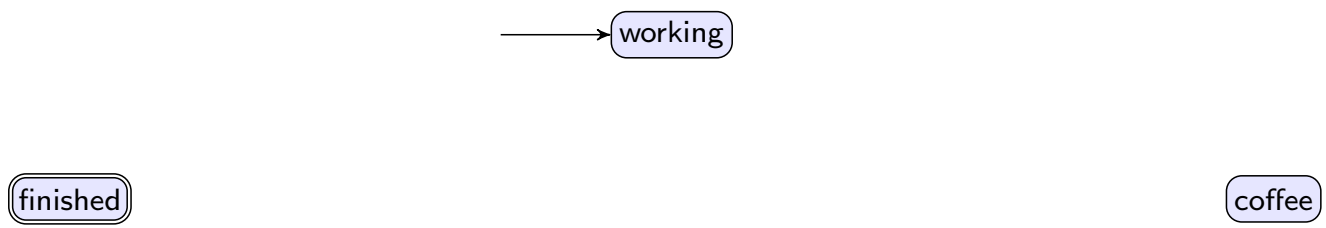
Yes if...

$$2 \times \text{delay} > 20.46 \times \text{period}$$

Outline

Timed automaton (TA)

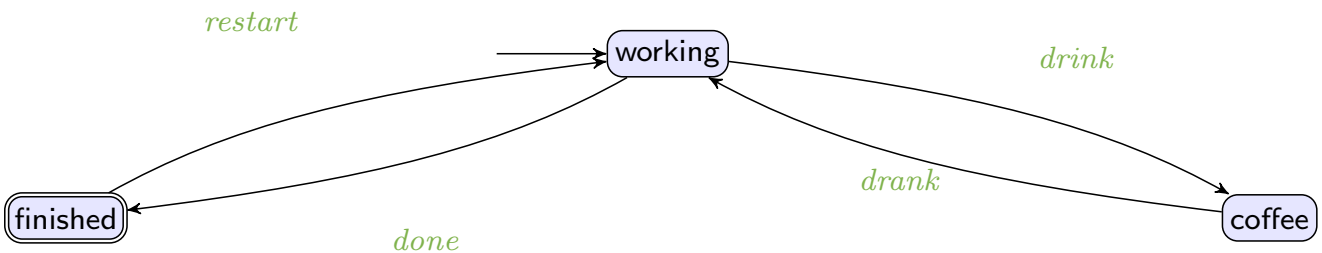
- Finite state automaton (sets of locations)



[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)

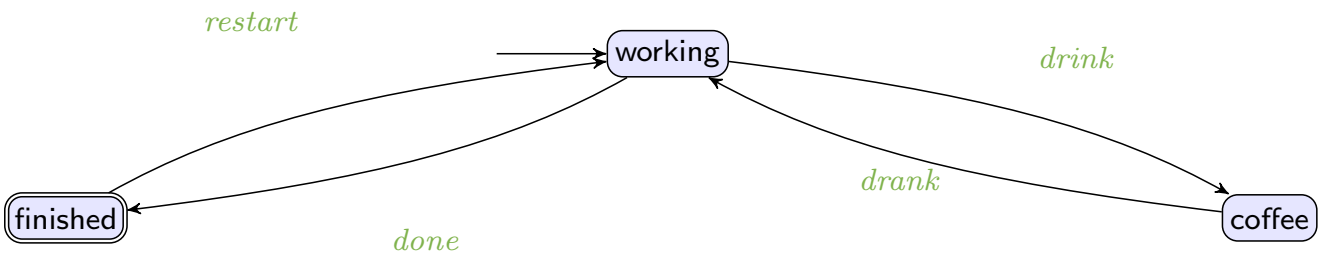
- Finite state automaton (sets of locations and actions)



[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
  - Can be compared to integer constants

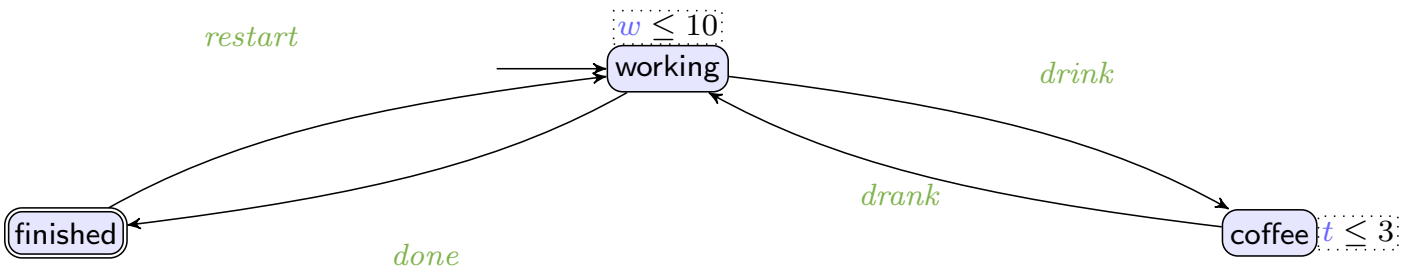


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

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

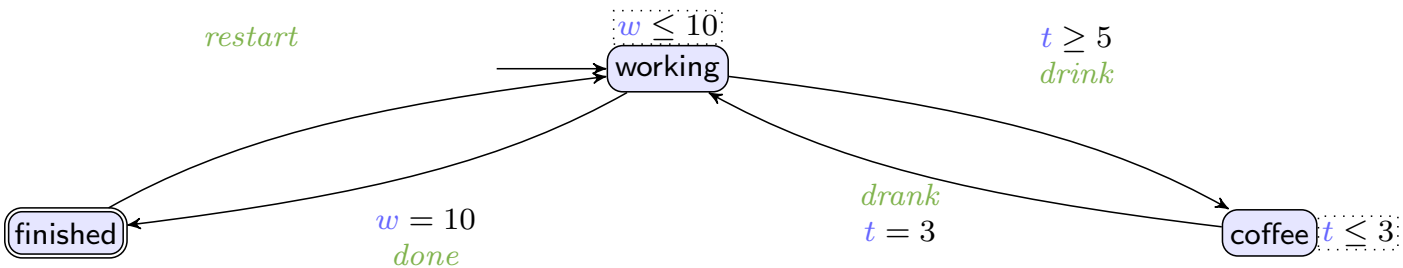


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

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

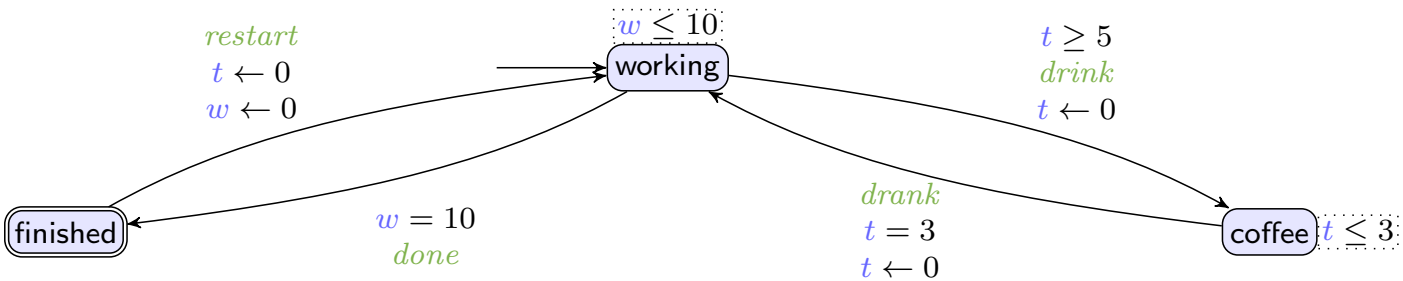


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

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

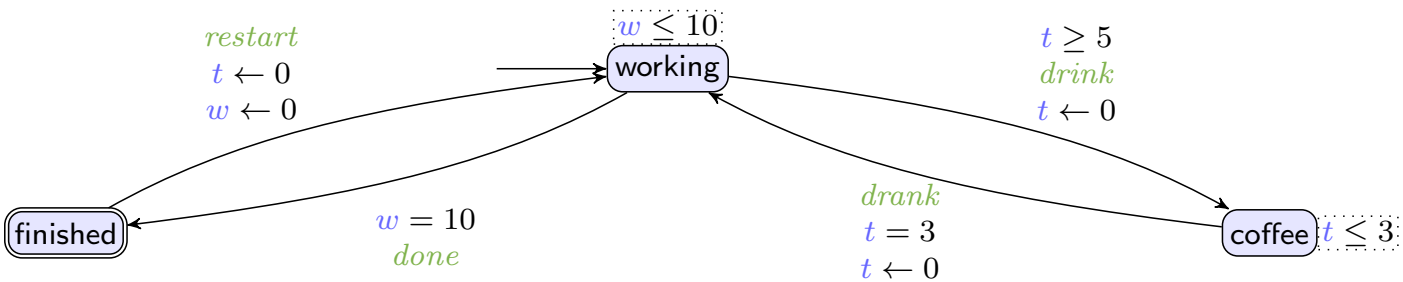


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235

Timed 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

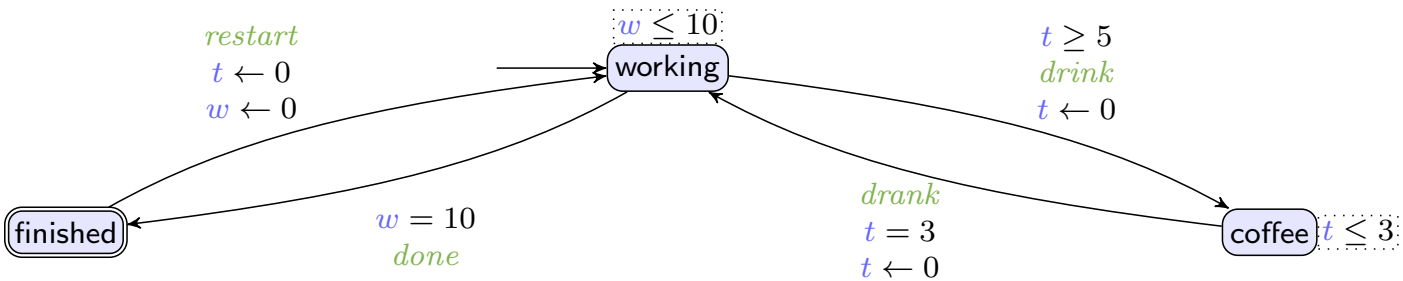


- Clock t: measuring the coffee time
- Clock w: measuring the amount of work done

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

Parametric Timed Automaton (PTA)

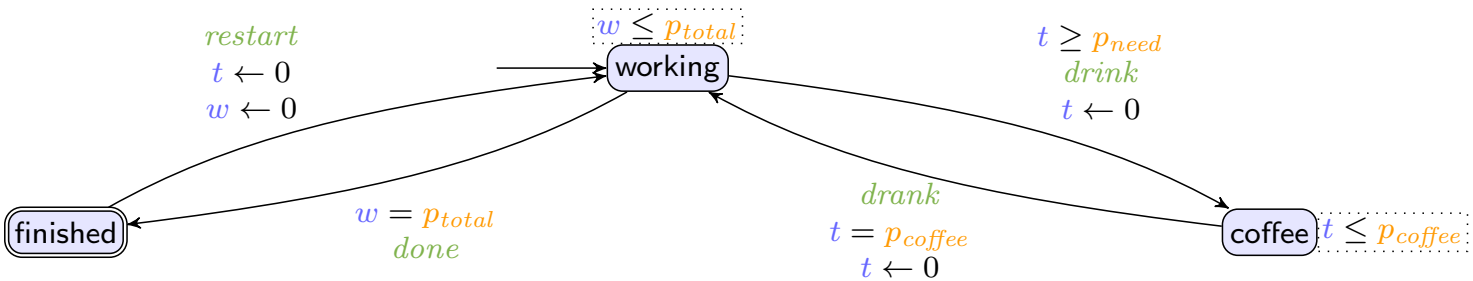
- Timed automaton (sets of locations, actions and clocks)



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

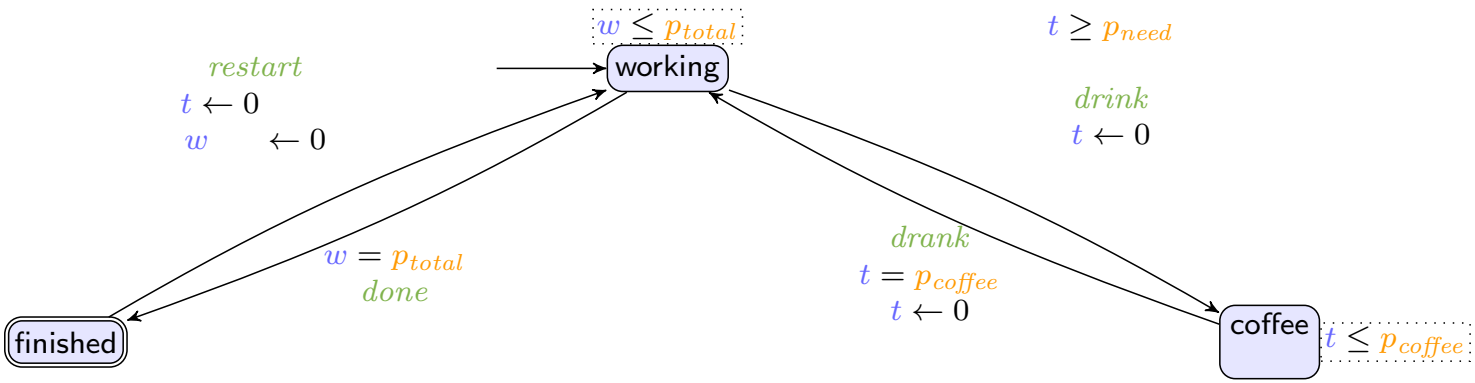
Parametric Timed Automaton (PTA)

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



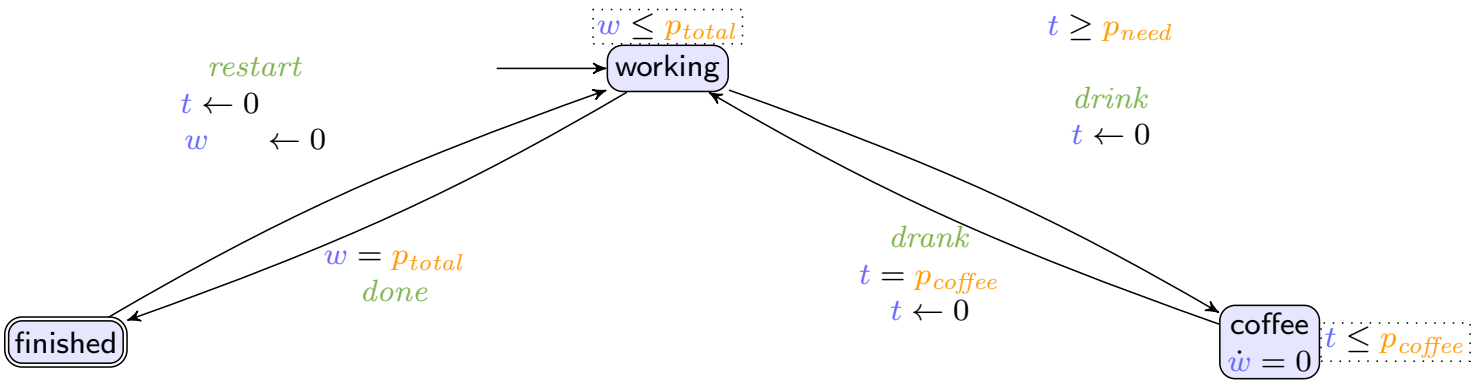
[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

IMITATOR 3 input formalism: a rich syntax



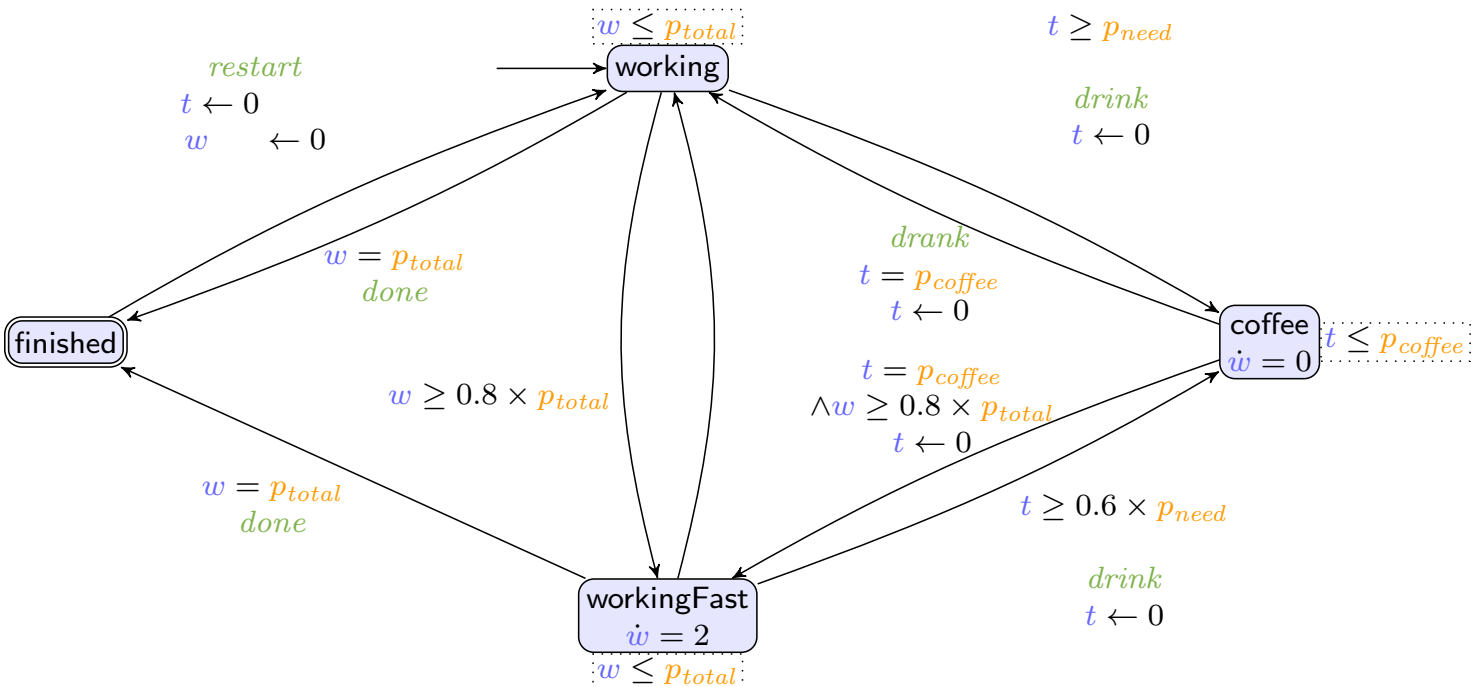
IMITATOR 3 input formalism: a rich syntax

- Stopwatches (stopping clock elapsing)



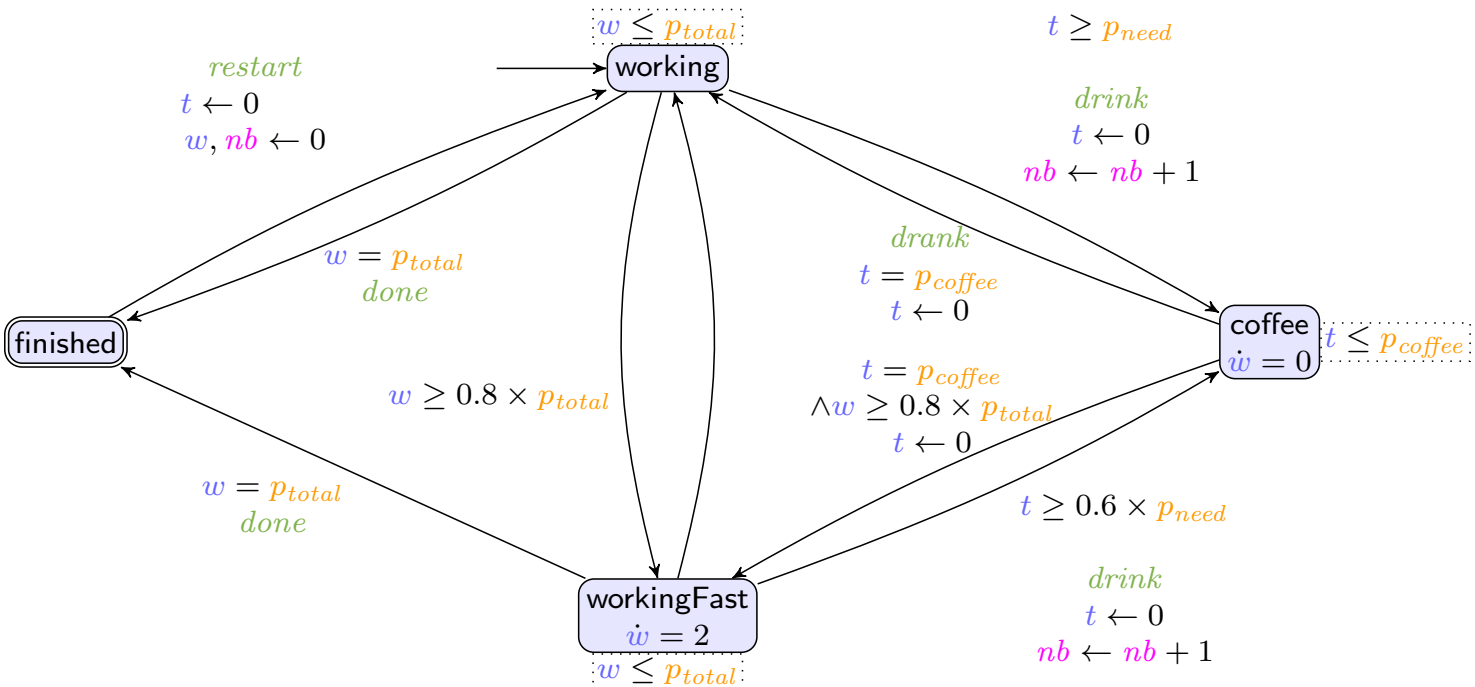
IMITATOR 3 input formalism: a rich syntax

- Stopwatches (stopping clock elapsing)
- Multi-rate variables (evolving at different speeds)



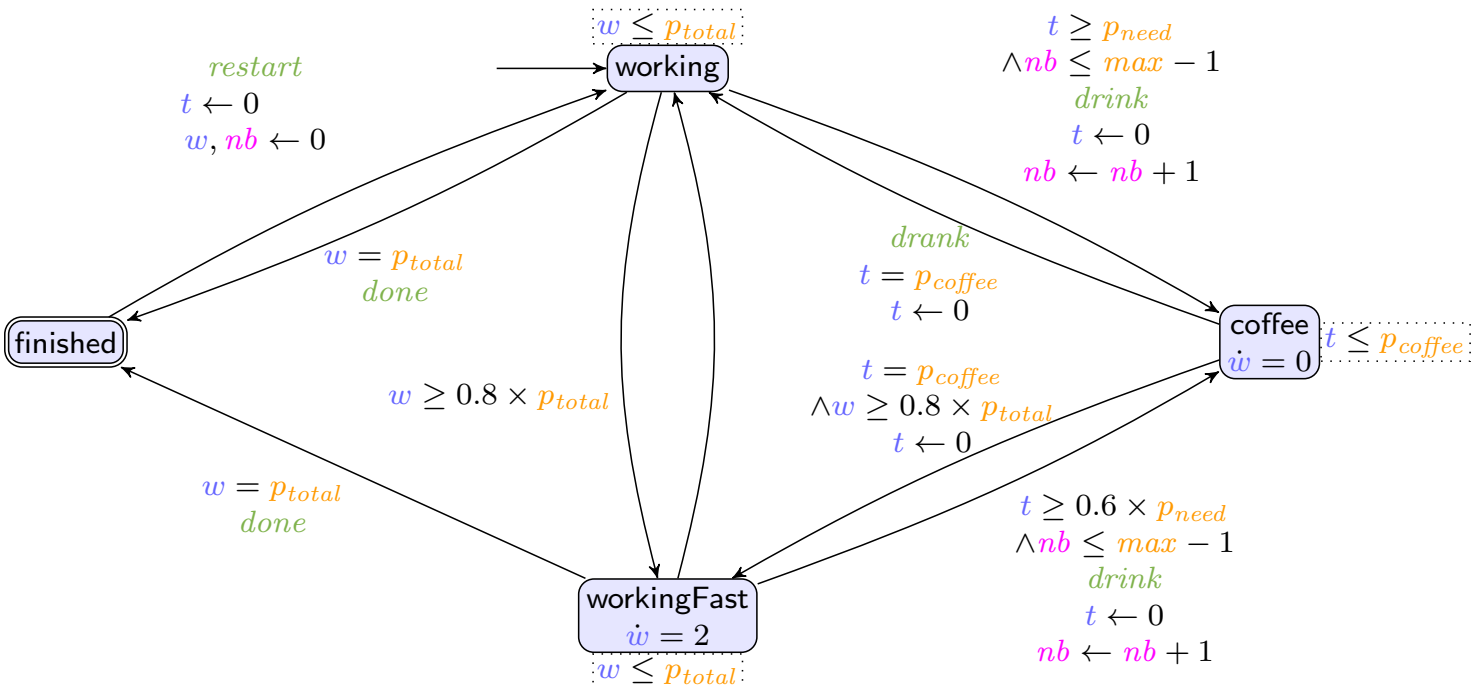
IMITATOR 3 input formalism: a rich syntax

- Stopwatches (stopping clock elapsing)
- Multi-rate variables (evolving at different speeds)
- Discrete rational variables (unbounded, exact arithmetics)



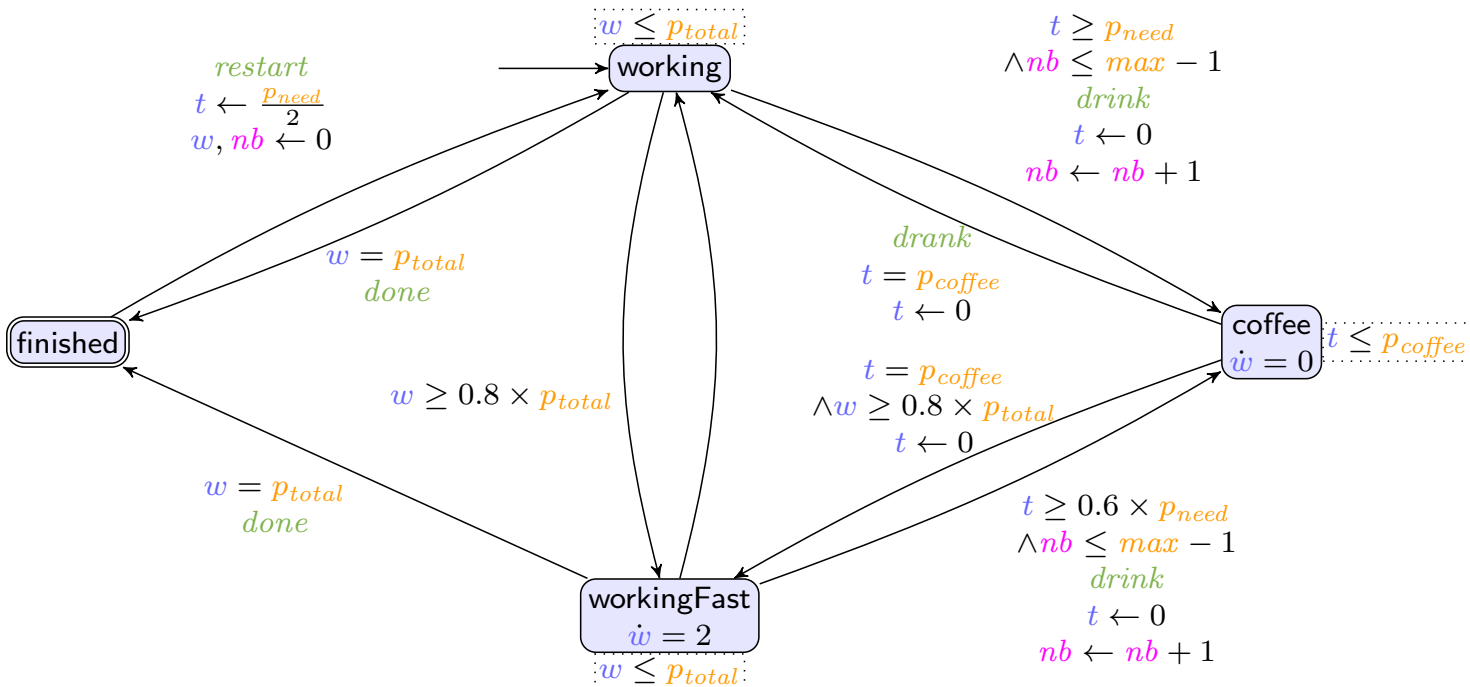
IMITATOR 3 input formalism: a rich syntax

- Stopwatches (stopping clock elapsing)
- Multi-rate variables (evolving at different speeds)
- Discrete rational variables (unbounded, exact arithmetics)
- Discrete parameters



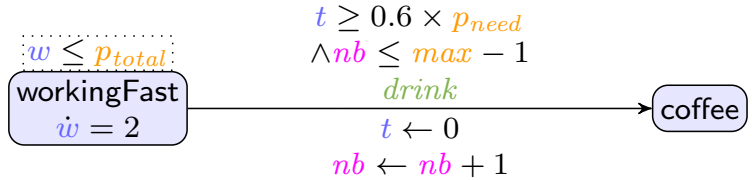
IMITATOR 3 input formalism: a rich syntax

- Stopwatches (stopping clock elapsing)
- Multi-rate variables (evolving at different speeds)
- Discrete rational variables (unbounded, exact arithmetics)
- Discrete parameters
- Parametric resets



Input syntax

- Text-based (originally inspired by HyTech)
- Human-friendly




```
loc workingFast: invariant w <= pTotal flow{w' = 2}
  when t >= 0.6 * pNeed & nb <= max - 1 sync drink do {t := 0,
     nb := nb + 1} goto coffee;
```

- Conversions to other formats
  - UPPAAL [LPY97] (losing parameters!)
  - JANI [Bud+17]
    - A new interchange format for automata-based formalisms

[LPY97] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. "UPPAAL in a Nutshell". In: *International Journal on Software Tools for Technology Transfer* 1.1-2 (1997), pp. 134–152

[Bud+17] Carlos E. Budde, Christian Dehnert, Ernst Moritz Hahn, Arnd Hartmanns, Sebastian Junges, and Andrea Turrini. "JANI: Quantitative Model and Tool Interaction". In: *TACAS*. vol. 10206. LNCS. 2017, pp. 151–168

Beyond decidability

- ☺ Timed automata benefit from (some) decidability results

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

[CL00] Franck Cassez and Kim Guldstrand Larsen. "The Impressive Power of Stopwatches". In: *CONCUR*. vol. 1877. LNCS. Springer, 2000, pp. 138–152

Beyond decidability

- ☺ Timed automata benefit from (some) decidability results
- ☹ Adding parameters yields undecidability [AHV93]
- ☹ Adding stopwatches yields undecidability [CL00]
- ☹ Adding unbounded rational variables yields undecidability

---

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

[CL00] Franck Cassez and Kim Guldstrand Larsen. "The Impressive Power of Stopwatches". In: *CONCUR*. vol. 1877. LNCS. Springer, 2000, pp. 138–152

Beyond decidability

- ☺ Timed automata benefit from (some) decidability results
- ☹ Adding parameters yields undecidability [AHV93]
- ☹ Adding stopwatches yields undecidability [CL00]
- ☹ Adding unbounded rational variables yields undecidability

IMITATOR paradigm: "best effort"

Try to synthesize parameter valuations

- No guarantee of termination, or
- Under or over-approximations and inform the user about them
  - Evaluate whether a result is exact, over-approximated, under-approximated, or possibly invalid

---

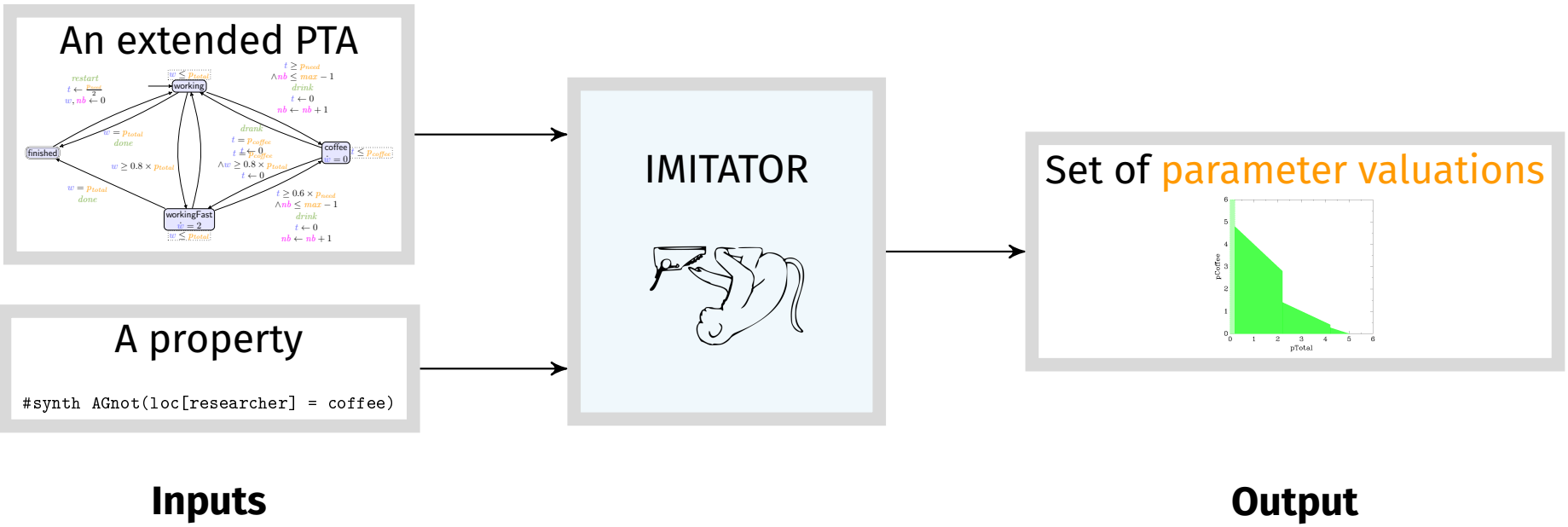
[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ACM, 1993, pp. 592–601

[CL00] Franck Cassez and Kim Guldstrand Larsen. "The Impressive Power of Stopwatches". In: *CONCUR*. vol. 1877. LNCS. Springer, 2000, pp. 138–152

Outline

Parameter synthesis using IMITATOR 3

IMITATOR is now a parametric timed model checker

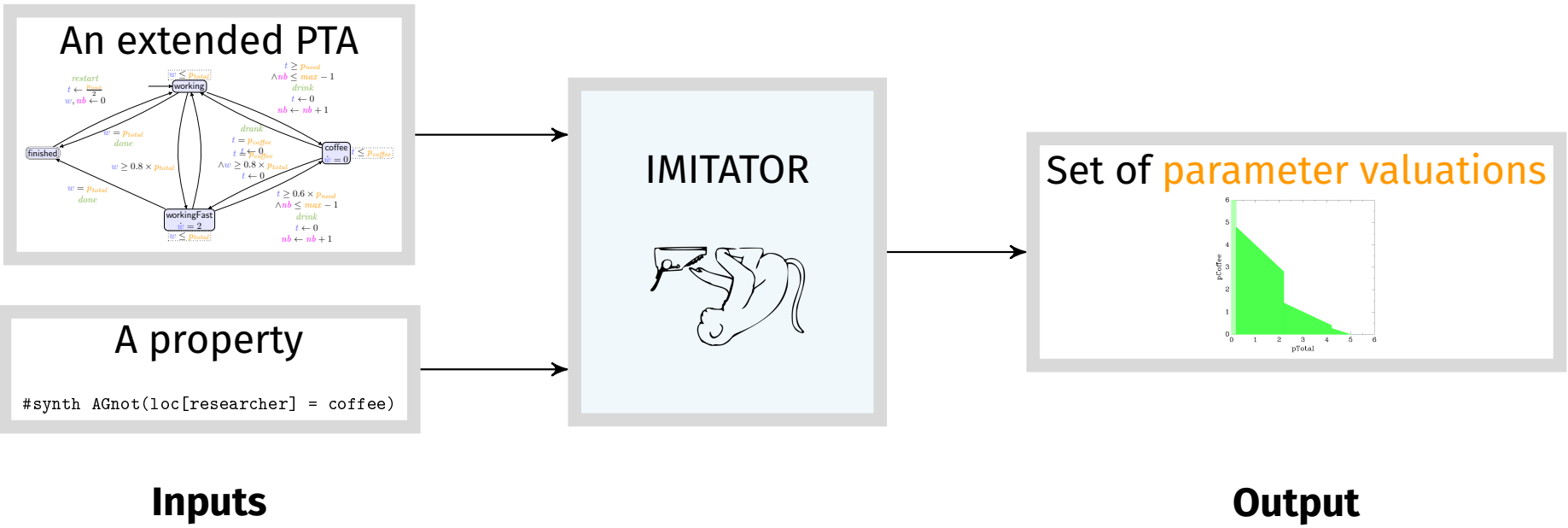


The set of parameter valuations is symbolic

- Symbolic: finite set of linear constraints (polyhedra)

Parameter synthesis using IMITATOR 3

IMITATOR is now a parametric timed model checker



The set of parameter valuations is symbolic

- Symbolic: finite set of linear constraints (polyhedra)
- Two categories of properties
  - Synthesis: “(try to) synthesize all valuations for which the property holds”
  - Exhibition: “(try to) synthesize at least one valuation for which the property holds”

Safety

Synthesize all parameter valuations for which the following property holds:

"It is impossible to drink any coffee"

(i. e., the coffee location is unreachable)

```
#synth AGnot(loc[researcher] = coffee)
```

Safety

Synthesize all parameter valuations for which the following property holds:
"It is impossible to drink any coffee"
(i. e., the coffee location is unreachable)

```
#synth AGnot(loc[researcher] = coffee)
```

Result:

$$max \in [0, 1) \vee \left(max \geq 1 \wedge p_{total} < \frac{p_{need}}{10}\right)$$

Safety: full result

```
(*************************************************************
 * Result by: IMITATOR 3.0 "Cheese" (build HEAD/ea560fd)
 * Model    : 'researcher.imi'
 * Generated: Mon Feb 1, 2021 14:57:17
 * Command  : imitator3 researcher.imi researcher-AGnotcoffee.imiprop
 *************************************************************)

BEGIN CONSTRAINT
 pTotal >= 0
& pNeed >= 1
& MAXBREAK >= 0
& pCoffee >= 0
& 1 > MAXBREAK
OR
  pNeed > 10*pTotal
& pTotal >= 0
& pNeed >= 1
& MAXBREAK >= 1
& pCoffee >= 0
END CONSTRAINT

------------------------------------------------------------
Constraint soundness                    : exact
Termination                             : regular termination
------------------------------------------------------------
```

Property patterns

IMITATOR 3 offers a set of predefined property patterns

- Simple, non-compositional, commonly met
- On the system actions and parameters
- Reduce to safety or reachability synthesis
- Also called observer patterns / reachability testing [Ace+03]

[Ace+03] Luca Aceto, Patricia Bouyer, Augusto Burgueño, and Kim Guldstrand Larsen. "The power of reachability testing for timed automata". In: *Theoretical Computer Science* 300.1-3 (2003), pp. 411–475

Property patterns

IMITATOR 3 offers a set of predefined property patterns

- Simple, non-compositional, commonly met
- On the system actions and parameters
- Reduce to safety or reachability synthesis
- Also called observer patterns / reachability testing [Ace+03]

```
#synth pattern(everytime restart then eventually done within 5)
```

[Ace+03] Luca Aceto, Patricia Bouyer, Augusto Burgueño, and Kim Guldstrand Larsen. "The power of reachability testing for timed automata". In: *Theoretical Computer Science* 300.1-3 (2003), pp. 411–475

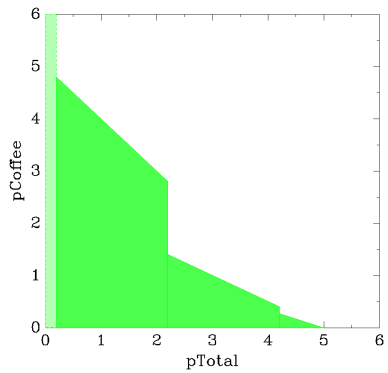
Property patterns

IMITATOR 3 offers a set of predefined property patterns

- Simple, non-compositional, commonly met
- On the system actions and parameters
- Reduce to safety or reachability synthesis
- Also called observer patterns / reachability testing [Ace+03]

```
#synth pattern(everytime restart then eventually done within 5)
```

Result projected onto 2 parameter dimensions:



[Ace+03] Luca Aceto, Patricia Bouyer, Augusto Burgueño, and Kim Guldstrand Larsen. "The power of reachability testing for timed automata". In: *Theoretical Computer Science* 300.1-3 (2003), pp. 411–475

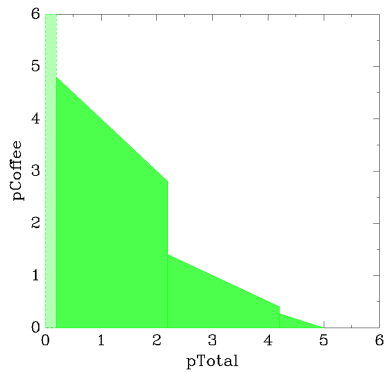
Property patterns

IMITATOR 3 offers a set of predefined property patterns

- Simple, non-compositional, commonly met
- On the system actions and parameters
- Reduce to safety or reachability synthesis
- Also called observer patterns / reachability testing [Ace+03]

```
#synth pattern(everytime restart then eventually done within 5)
```

Result projected onto 2 parameter dimensions:



IMITATOR patterns can be parameterized (e.g., `within p`)

[Ace+03] Luca Aceto, Patricia Bouyer, Augusto Burgueño, and Kim Guldstrand Larsen. "The power of reachability testing for timed automata". In: *Theoretical Computer Science* 300.1-3 (2003), pp. 411–475

Optimal parameter reachability

Goal: synthesizing valuations for which the value of a given parameter is minimized or maximized when reaching a given state

Example: synthesize the valuations minimizing the value of p_{total} when finishing a paper after drinking (at least) 3 coffees

```
#synth EFpmin(loc[researcher] = finished & nb >= 3, pTotal)
```

Optimal parameter reachability

Goal: synthesizing valuations for which the value of a given parameter is minimized or maximized when reaching a given state

Example: synthesize the valuations minimizing the value of p_{total} when finishing a paper after drinking (at least) 3 coffees

```
#synth EFpmin(loc[researcher] = finished & nb >= 3, pTotal)
```

Result:

$$max \geq 3 \wedge p_{total} = 2.1 \wedge p_{need} = 1$$

- Note: p_{coffee} is not involved in this constraint: the time spent in drinking coffee does not impact the total duration of the *work* (p_{total}), as the progress of clock x is stopped in coffee

Liveness

Büchi acceptance condition

Example: valuations for which there exists a run s.t. the researcher completes a paper infinitely often

```
#synth CycleThrough(loc[researcher] = finished)
```

[NPP18] Hoang Gia Nguyen, Laure Petrucci, and Jaco van de Pol. "Layered and Collecting NDFS with Subsumption for Parametric Timed Automata". In: *ICECCS*. IEEE Computer Society, Dec. 2018, pp. 1–9

[And+21] Étienne André, Jaime Arias, Laure Petrucci, and Jaco van de Pol. "Iterative Bounded Synthesis for Efficient Cycle Detection in Parametric Timed Automata". In: *TACAS*. vol. 12651. Springer, 2021, pp. 311–329

Liveness

Büchi acceptance condition

Example: valuations for which there exists a run s.t. the researcher completes a paper infinitely often

```
#synth CycleThrough(loc[researcher] = finished)
```

Result: True (i. e., all valuations may yield such behavior)

[NPP18] Hoang Gia Nguyen, Laure Petrucci, and Jaco van de Pol. "Layered and Collecting NDFS with Subsumption for Parametric Timed Automata". In: *ICECCS*. IEEE Computer Society, Dec. 2018, pp. 1–9

[And+21] Étienne André, Jaime Arias, Laure Petrucci, and Jaco van de Pol. "Iterative Bounded Synthesis for Efficient Cycle Detection in Parametric Timed Automata". In: *TACAS*. vol. 12651. Springer, 2021, pp. 311–329

Liveness

Büchi acceptance condition

Example: valuations for which there exists a run s.t. the researcher completes a paper infinitely often

```
#synth CycleThrough(loc[researcher] = finished)
```

Result: True (i. e., all valuations may yield such behavior)

In the box:

- Variants of BFS
- NDFS extended with parametric subsumption and pruning [NPP18][And+21]

[NPP18] Hoang Gia Nguyen, Laure Petrucci, and Jaco van de Pol. "Layered and Collecting NDFS with Subsumption for Parametric Timed Automata". In: *ICECCS*. IEEE Computer Society, Dec. 2018, pp. 1–9

[And+21] Étienne André, Jaime Arias, Laure Petrucci, and Jaco van de Pol. "Iterative Bounded Synthesis for Efficient Cycle Detection in Parametric Timed Automata". In: *TACAS*. vol. 12651. Springer, 2021, pp. 311–329

Trace preservation (robustness)

Quantifying the admissible variations of some parameters w.r.t. the discrete (untimed) behavior

```
#synth TracePreservation(pTotal=10, pNeed=5, pCoffee=3, max=3)
```

Trace preservation (robustness)

Quantifying the admissible variations of some parameters w.r.t. the discrete (untimed) behavior

```
#synth TracePreservation(pTotal=10, pNeed=5, pCoffee=3, max=3)
```

Result:

$$\left(3\times p_{need} > p_{total} \geq 2\times p_{need} \wedge max \in [2,3)\right) \vee \left(2.1\times p_{need} > p_{total} \geq 2\times p_{need} \wedge max \geq 3\right)$$

And also…

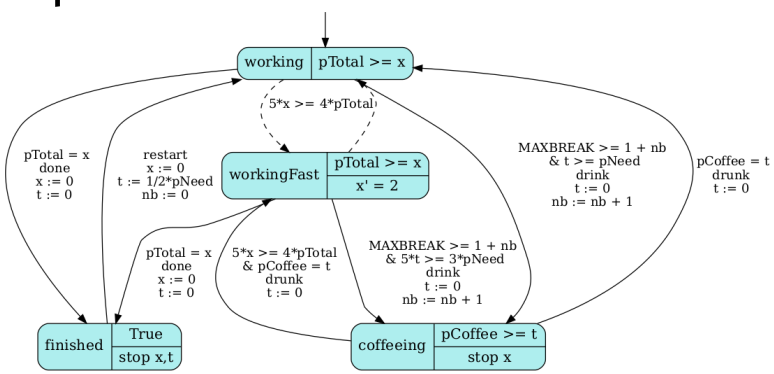
- Deadlock freeness
- Minimal-time reachability
- Parametric reachability preservation
- Behavioral cartography
- …

Results

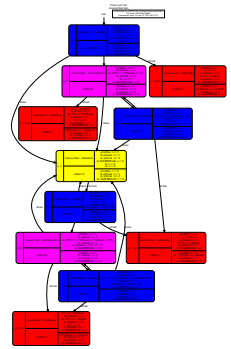
- Normalized text results

```
 * Result by: IMITATOR 3.0 "Cheese" (build HEAD/ea560fd)
 * Model     : 'researcher.imi'
 * Generated: Mon Feb 1, 2021 14:57:17
 * Command   : imitator3 researcher.imi researcher-AGnotcoffee.imiprop
 ************************************************************)

BEGIN CONSTRAINT
 pTotal >= 0
& pNeed >= 1
& MAXBREAK >= 0
& pCoffee >= 0
& 1 > MAXBREAK
OR
  pNeed > 10*pTotal
& pTotal >= 0
& pNeed >= 1
& MAXBREAK >= 1
& pCoffee >= 0
END CONSTRAINT

------------------------------------------------------------
Constraint soundness                    : exact
Termination                             : regular termination
------------------------------------------------------------
```

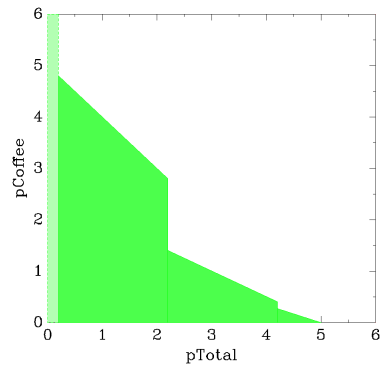
- Graphical results



PTA visualization



State space (zone graph)



Constraints representation

Outline

Under the box

Entirely written in OCaml

Strongly relies on polyhedra for symbolic computations

- Parma polyhedra library [BHZ08]

[BHZ08] Roberto Bagnara, Patricia M. Hill, and Enea Zaffanella. "The Parma Polyhedra Library: Toward a Complete Set of Numerical Abstractions for the Analysis and Verification of Hardware and Software Systems". In: *Science of Computer Programming* 72.1–2 (2008), pp. 3–21

Distribution

Free and open source software: Available under the GNU-GPL license

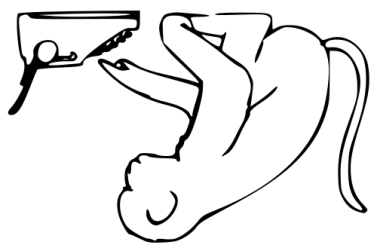
Distribution:

- Binaries available for Linux platforms (no dependency, no install)
- Docker version
- Integrated as a virtual machine

doi.org/10.5281/zenodo.4723415

- Comes with a user manual and an extensive benchmarks library

Try it!



www.imitator.fr

Outline

Some success stories

- Verification of an asynchronous memory circuit by ST-Microelectronics
- Parametric schedulability analyses for flight control systems for ASTRIUM Space Transportation / ArianeGroup [Fri+12]
- Verification of software product lines [Lut+17]
- Formal timing analysis of music scores [FJ13]
- Solution to a challenge related to a distributed video processing system by Thales
- Parametric timed pattern matching and online monitoring [AHW18]

[Fri+12] Laurent Fribourg, David Lesens, Pierre Moro, and Romain Soulat. "Robustness Analysis for Scheduling Problems using the Inverse Method". In: *TIME*. IEEE Computer Society Press, Sept. 2012, pp. 73–80

[Lut+17] Lars Luthmann, Andreas Stephan, Johannes Bürdek, and Malte Lochau. "Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints". In: *SPLC, Volume A*. ACM, 2017, pp. 104–113

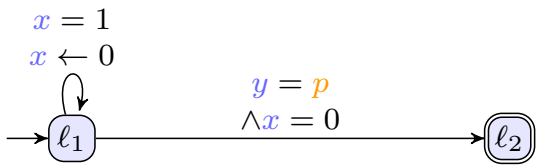
[FJ13] Léa Fanchon and Florent Jacquemard. "Formal Timing Analysis Of Mixed Music Scores". In: *ICMC*. Michigan Publishing, Aug. 2013

[AHW18] Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS*. IEEE Computer Society, 2018, pp. 10–20

Outline

Perspectives

- Solving problems not representable by a finite union of polyhedra
  - Toy benchmark for which the answer is $\{p = i, i \in \mathbb{N}\}$



- Discrete parameters (as in population protocols)
  - "Arbitrary number of concurrent coffee drinkers"

- Integration to higher-level formalisms
  - Logics: MITL, STL
  - Real-time systems

Thanks to the contributors!

Jaime Arias

Vincent Bloemen

Camille Coti

Sami Evangelista

Dylan Marinho

Nguyen Hoang Gia

Laure Petrucci

Jaco van de Pol

Bibliography

References I

Luca Aceto, Patricia Bouyer, Augusto Burgueño, and Kim Guldstrand Larsen. "The power of reachability testing for timed automata". In: *Theoretical Computer Science* 300.1-3 (2003), pp. 411–475. DOI: 10.1016/S0304-3975(02)00334-1.

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.

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, United States: ACM, 1993, pp. 592–601. DOI: 10.1145/167088.167242.

Étienne André, Ichiro Hasuo, and Masaki Waga. "Offline timed pattern matching under uncertainty". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, 2018, pp. 10–20. DOI: 10.1109/ICECCS2018.2018.00010.

Étienne André, Jaime Arias, Laure Petrucci, and Jaco van de Pol. "Iterative Bounded Synthesis for Efficient Cycle Detection in Parametric Timed Automata". In: *TACAS* (Mar. 27–Apr. 1, 2021). Ed. by Jan Friso Groote and Kim G. Larsen. Vol. 12651. Virtual: Springer, 2021, pp. 311–329. DOI: 10.1007/978-3-030-72016-2_17.

Étienne André. "IMITATOR 3: Synthesis of timing parameters beyond decidability". In: *CAV* (July 18–23, 2021). Ed. by Rustan Leino and Alexandra Silva. virtual, 2021.

References II

Roberto Bagnara, Patricia M. Hill, and Enea Zaffanella. "The Parma Polyhedra Library: Toward a Complete Set of Numerical Abstractions for the Analysis and Verification of Hardware and Software Systems". In: *Science of Computer Programming* 72.1–2 (2008), pp. 3–21. DOI: 10.1016/j.scico.2007.08.001.

Patricia Bouyer, Nicolas Markey, and Ocan Sankur. "Robustness in timed automata". In: *RP*. Ed. by Parosh Aziz Abdulla and Igor Potapov. Vol. 8169. LNCS. Invited paper. Uppsala, Sweden: Springer, Sept. 2013, pp. 1–18. DOI: 10.1007/978-3-642-41036-9_1.

Carlos E. Budde, Christian Dehnert, Ernst Moritz Hahn, Arnd Hartmanns, Sebastian Junges, and Andrea Turrini. "JANI: Quantitative Model and Tool Interaction". In: *TACAS* (Apr. 22–29, 2017). Ed. by Axel Legay and Tiziana Margaria. Vol. 10206. LNCS. Uppsala, Sweden, 2017, pp. 151–168. DOI: 10.1007/978-3-662-54580-5_9.

Franck Cassez and Kim Guldstrand Larsen. "The Impressive Power of Stopwatches". In: *CONCUR* (Aug. 22–25, 2000). Ed. by Catuscia Palamidessi. Vol. 1877. LNCS. University Park, PA, USA: Springer, 2000, pp. 138–152. DOI: 10.1007/3-540-44618-4_12.

Léa Fanchon and Florent Jacquemard. "Formal Timing Analysis Of Mixed Music Scores". In: *ICMC* (Aug. 12–16, 2013). Perth, Australia: Michigan Publishing, Aug. 2013.

References III

Laurent Fribourg, David Lesens, Pierre Moro, and Romain Soulat. "Robustness Analysis for Scheduling Problems using the Inverse Method". In: *TIME* (Sept. 12–14, 2012). Ed. by Mark Reynolds, Paolo Terenziani, and Ben Moszkowski. Leicester, UK: IEEE Computer Society Press, Sept. 2012, pp. 73–80. DOI: 10.1109/TIME.2012.10.

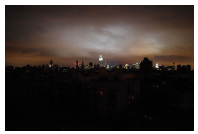
Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. "UPPAAL in a Nutshell". In: *International Journal on Software Tools for Technology Transfer* 1.1-2 (1997), pp. 134–152. DOI: 10.1007/s100090050010.

Lars Luthmann, Andreas Stephan, Johannes Bürdek, and Malte Lochau. "Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints". In: *SPLC, Volume A* (Sept. 25–29, 2017). Ed. by Myra B. Cohen, Mathieu Acher, Lidia Fuentes, Daniel Schall, Jan Bosch, Rafael Capilla, Ebrahim Bagheri, Yingfei Xiong, Javier Troya, Antonio Ruiz Cortés, and David Benavides. Sevilla, Spain: ACM, 2017, pp. 104–113. DOI: 10.1145/3106195.3106204.

Hoang Gia Nguyen, Laure Petrucci, and Jaco van de Pol. "Layered and Collecting NDFS with Subsumption for Parametric Timed Automata". In: *ICECCS* (Dec. 12–14, 2018). Ed. by Anthony Widjaja Lin and Jun Sun. Melbourne, Australia: IEEE Computer Society, Dec. 2018, pp. 1–9. DOI: 10.1109/ICECCS2018.2018.00009.

Additional information

Explanation for the 3 pictures in the beginning



Allusion to the Northeast blackout (USA, 2003)
Computer bug
Consequences: 11 fatalities, huge cost
(Picture actually from the Sandy Hurricane, 2012)

Allusion to the MIM-104 Patriot Missile Failure (Iraq, 1991)
28 fatalities, hundreds of injured
Computer bug: software error (clock drift)
(Picture of an actual MIM-104 Patriot Missile, though not the one of 1991)

Allusion to the sinking of the Sleipner A offshore platform (Norway, 1991)
No fatalities
Computer bug: inaccurate finite element analysis modeling
(Picture actually from the Deepwater Horizon Offshore Drilling Platform)

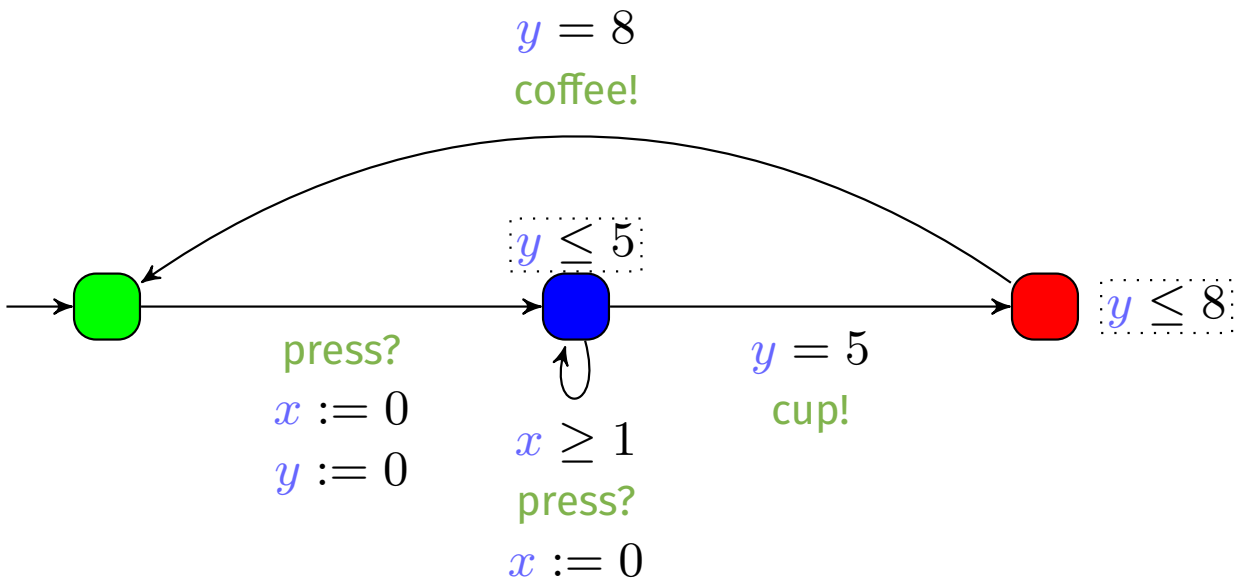
Concrete semantics of timed automata

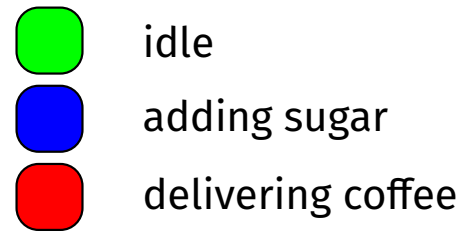
- Concrete state of a TA: pair (ℓ, w), where
  - ℓ is a location,
  - w is a valuation of each clock

  Example: $\left(\right.$  $, \begin{pmatrix} x=1.2 \\ y=3.7 \end{pmatrix}\left.\right)$

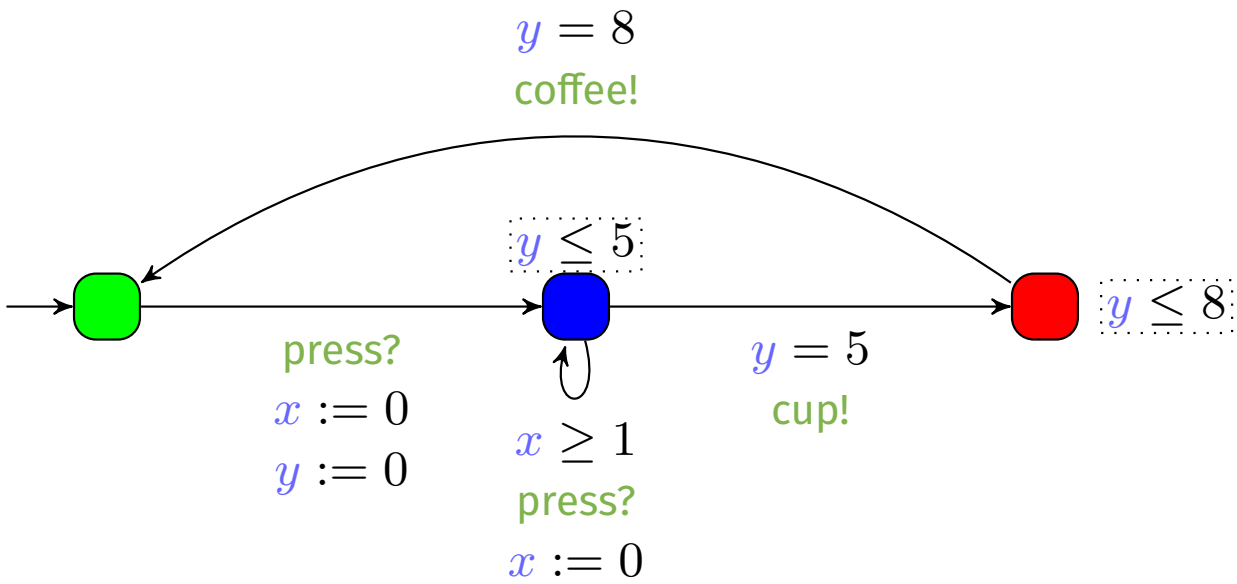
- Concrete run: alternating sequence of concrete states and actions or time elapse

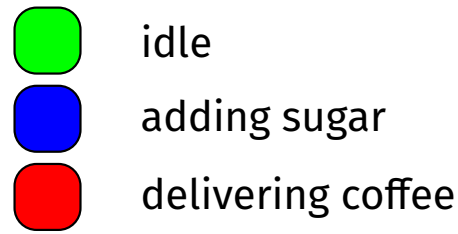
An example of TA concrete run



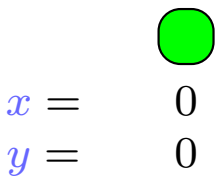


An example of TA concrete run

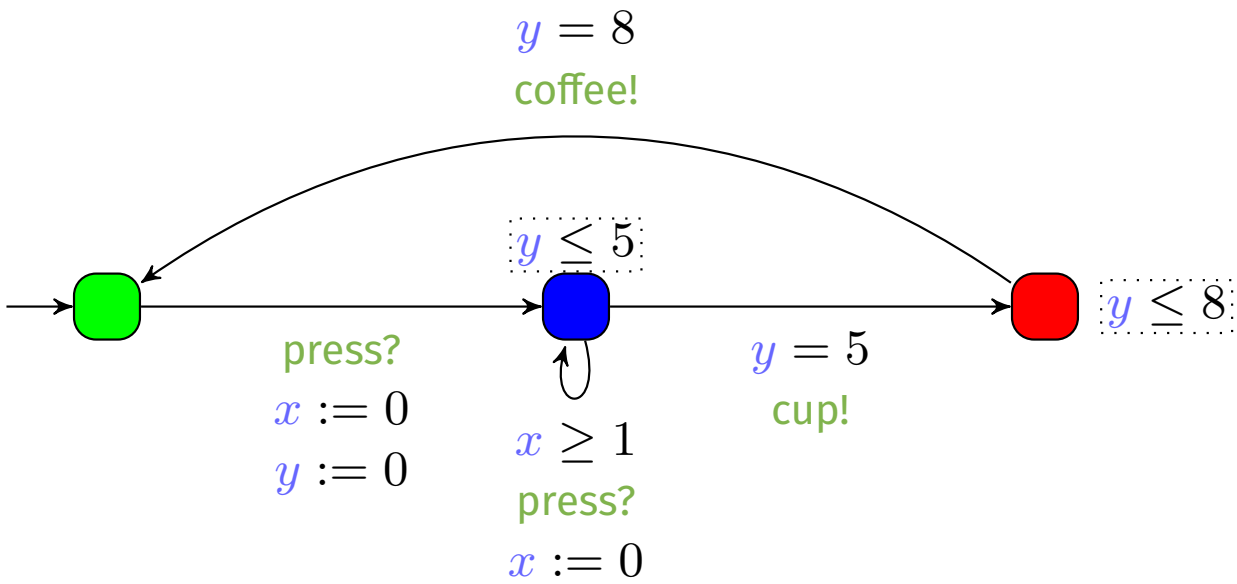


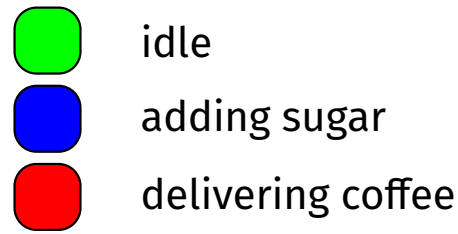


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

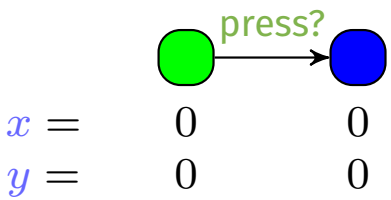


An example of TA concrete run

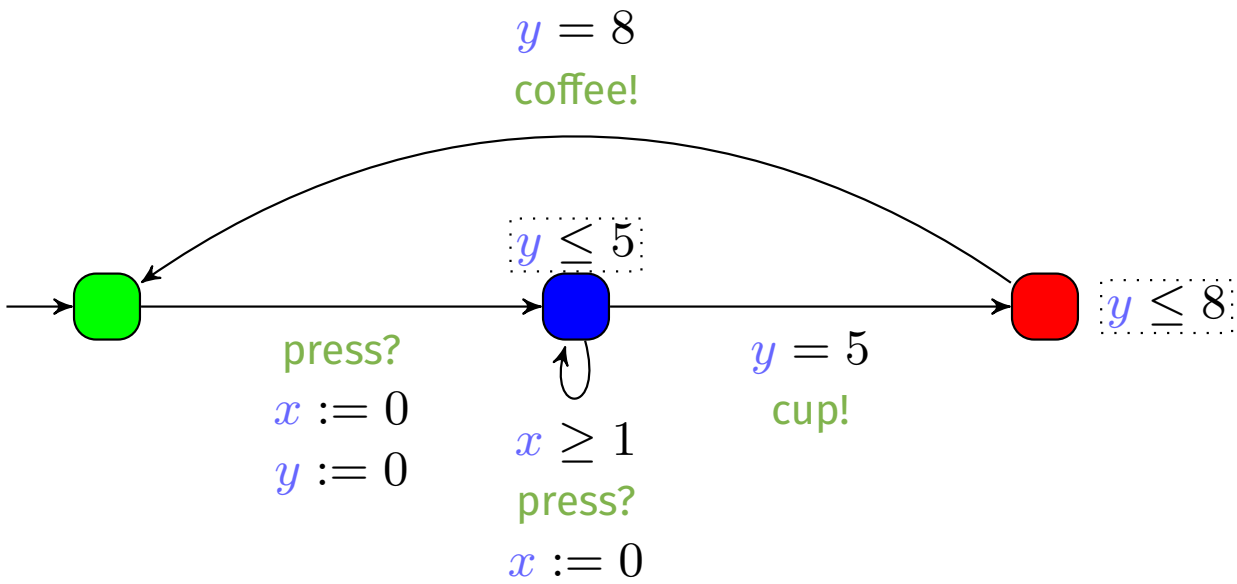


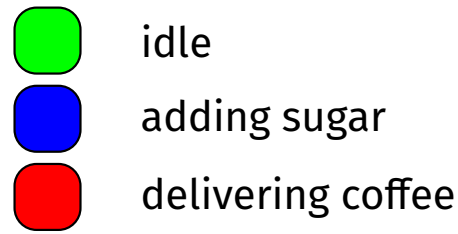


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

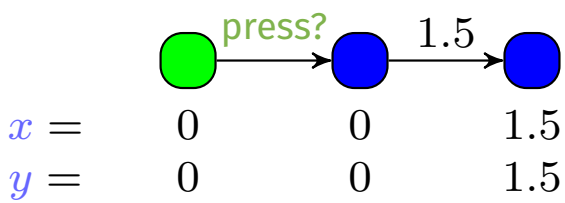


An example of TA concrete run

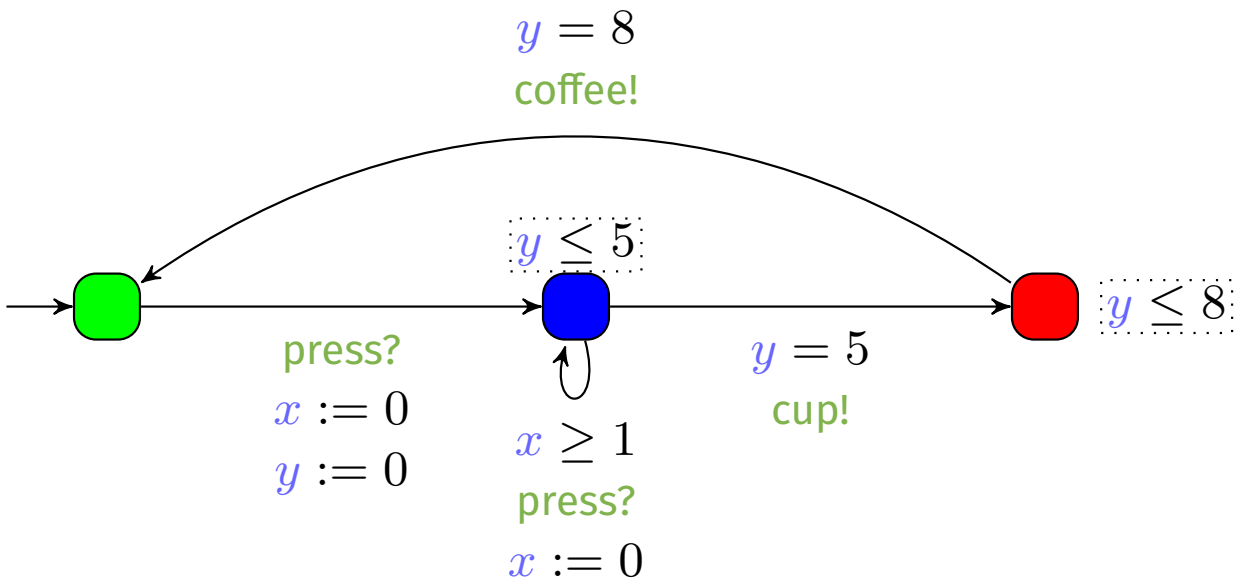


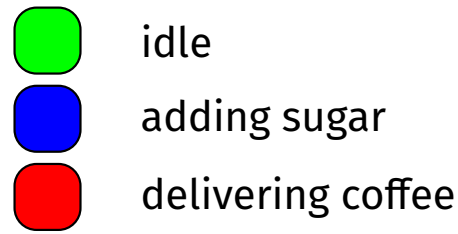


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

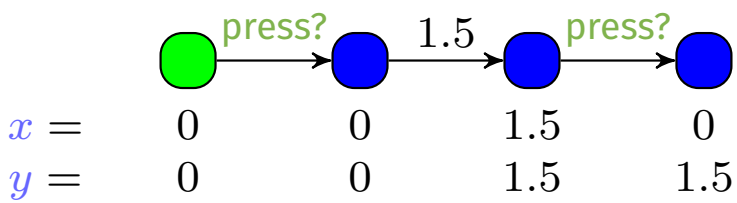


An example of TA concrete run

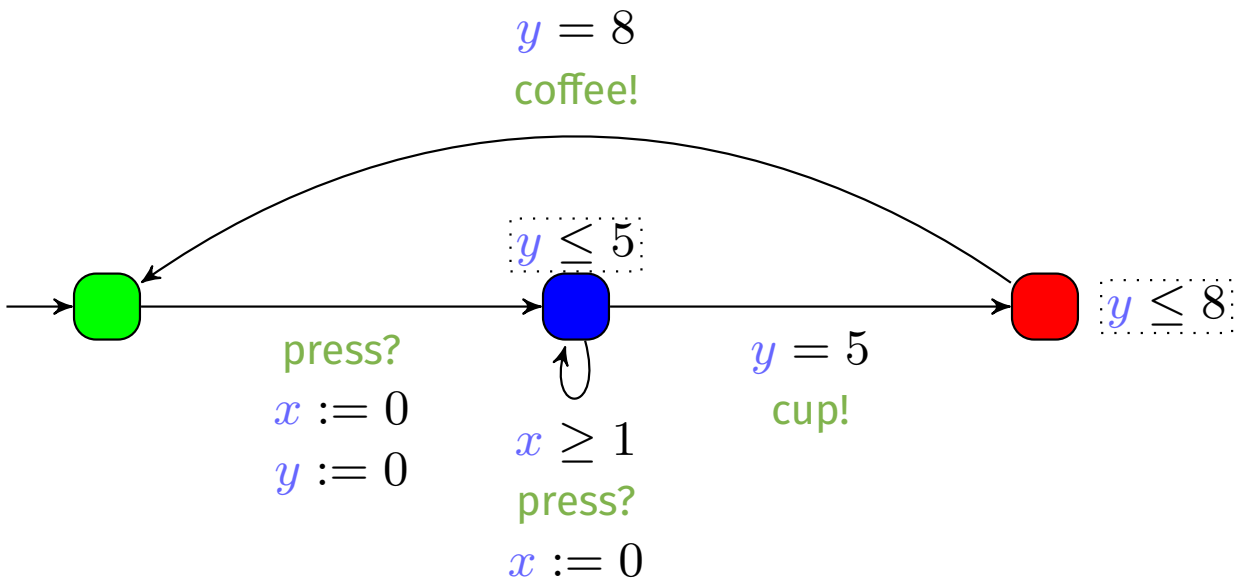


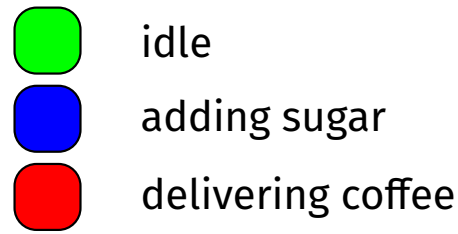


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

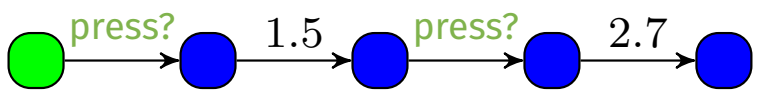


An example of TA concrete run



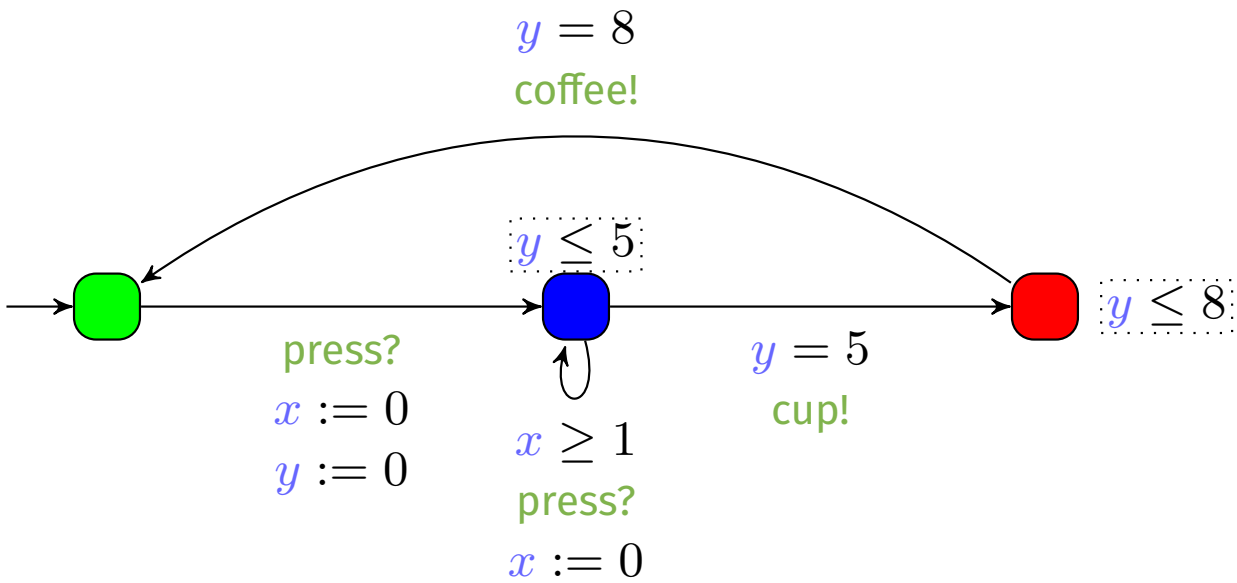


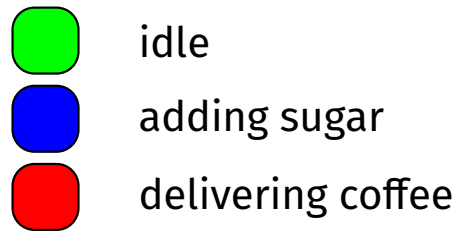
- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



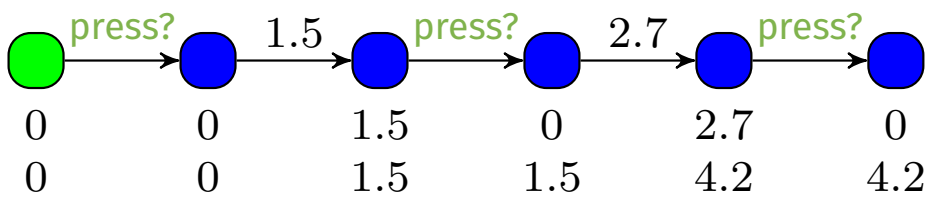
$x =$	0	0	1.5	0	2.7
$y =$	0	0	1.5	1.5	4.2

An example of TA concrete run

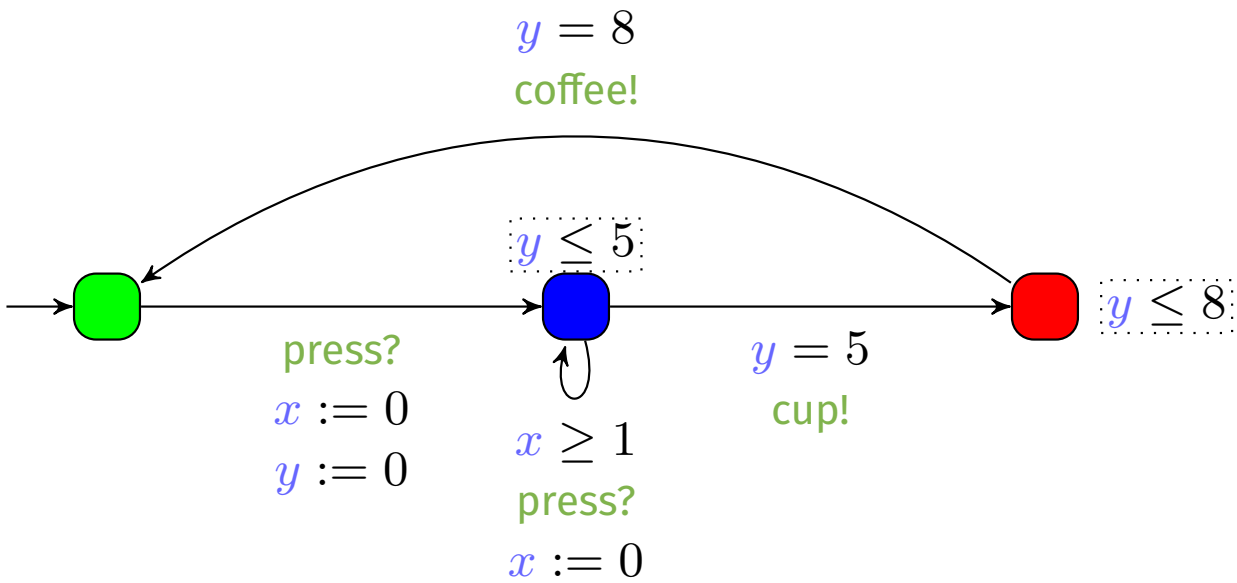


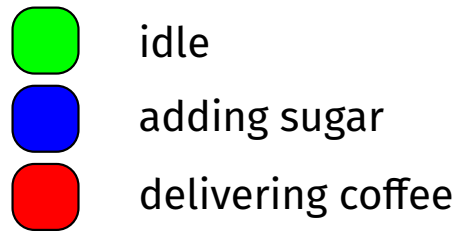


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

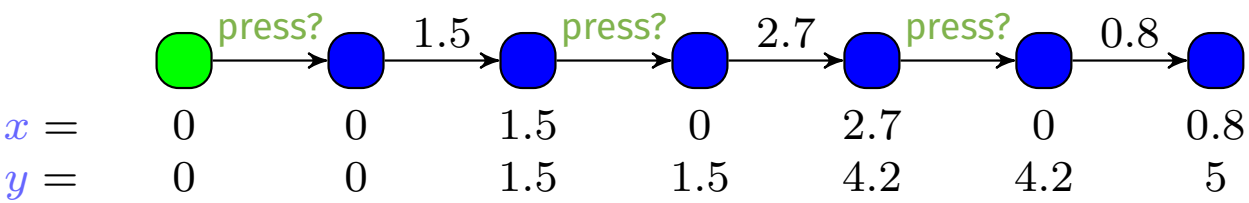


An example of TA concrete run

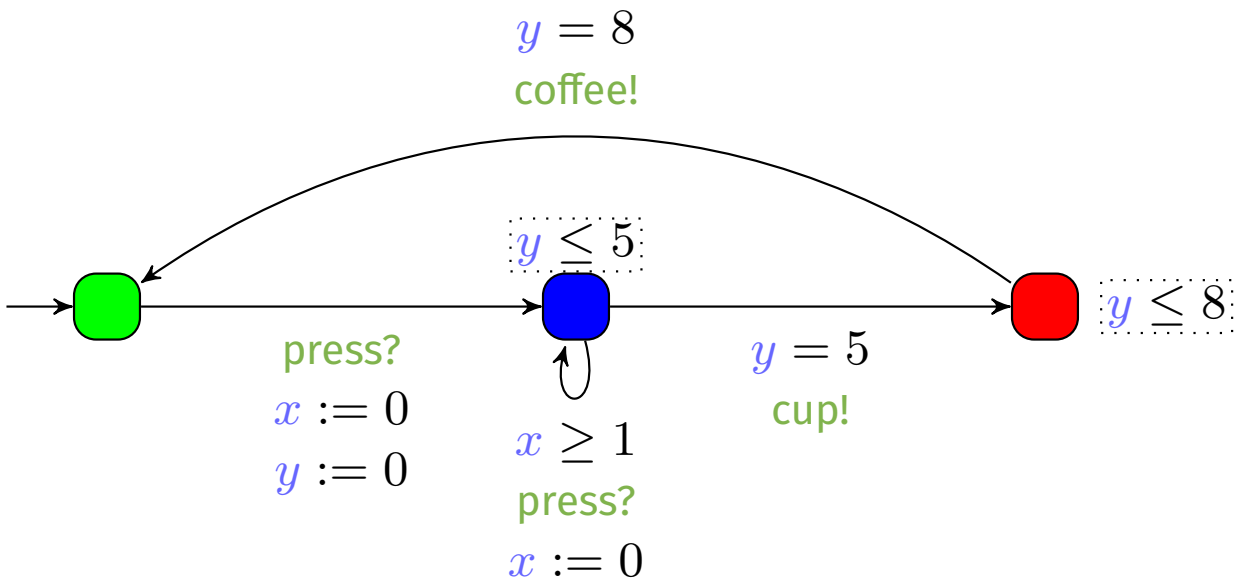


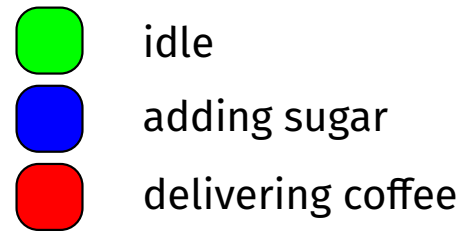


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

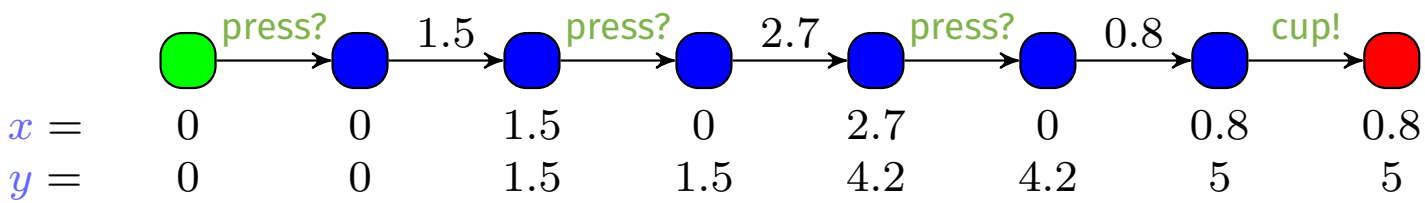


An example of TA concrete run

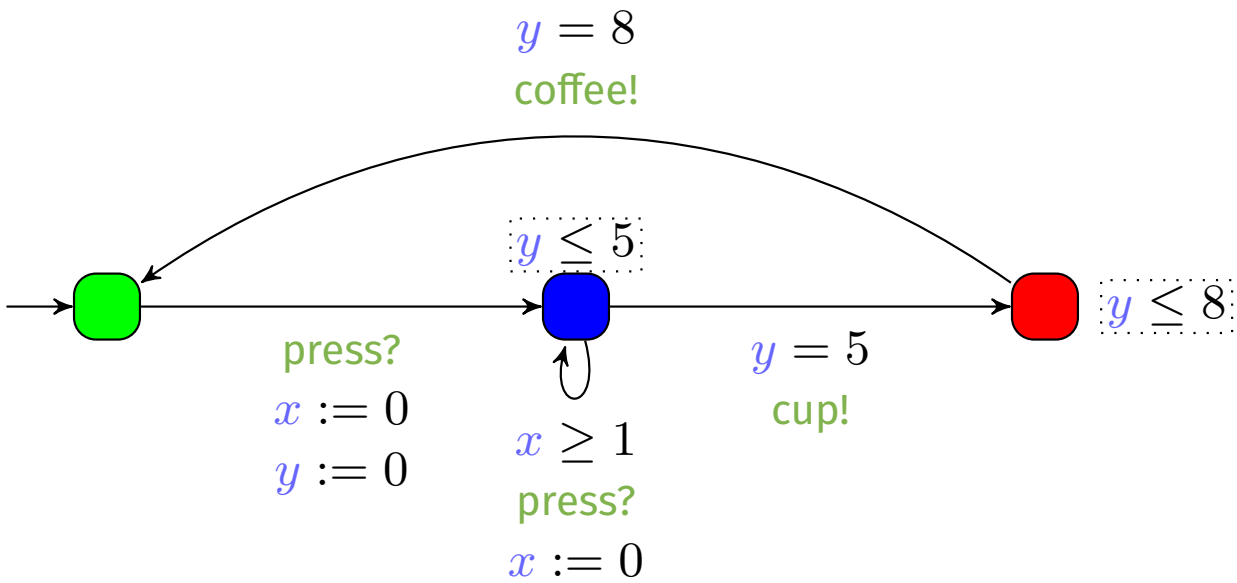


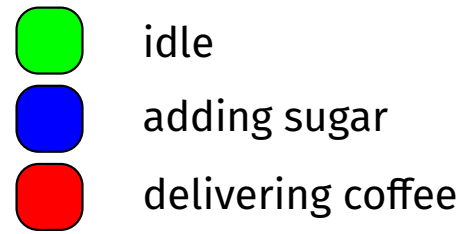


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

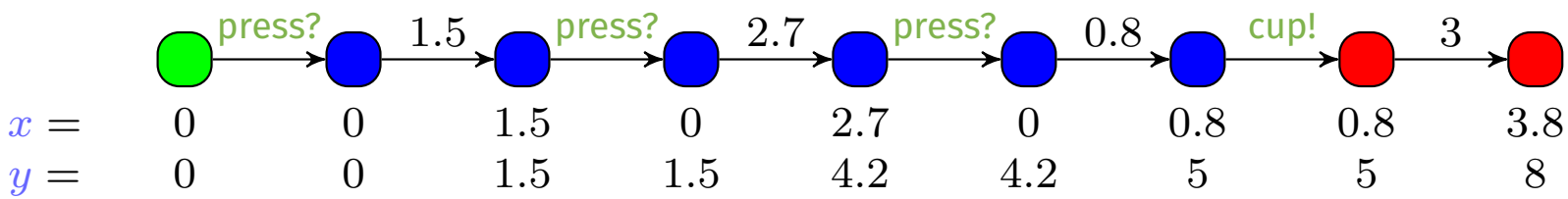


An example of TA concrete run

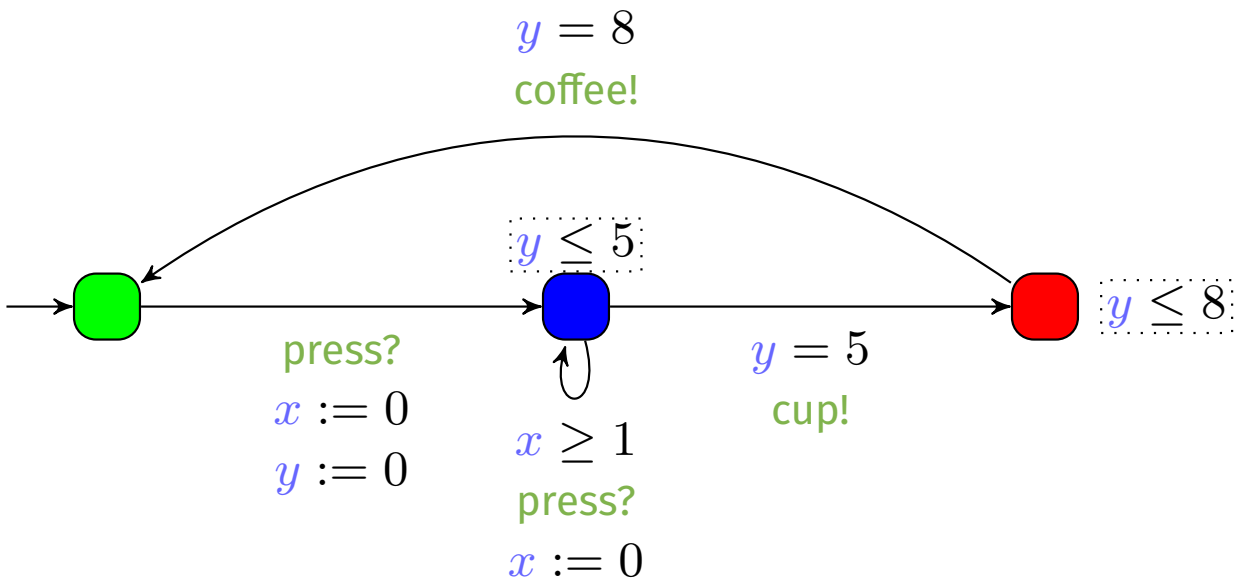


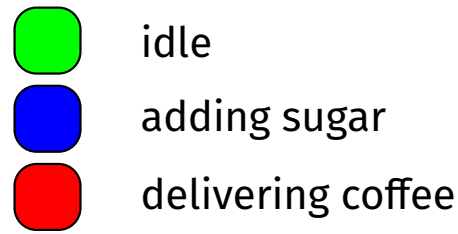


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar

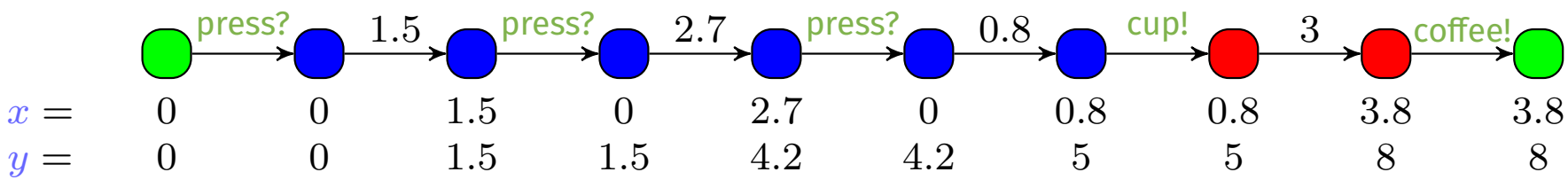


An example of TA concrete run



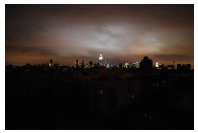


- Example of concrete run for the coffee machine
  - Coffee with 2 doses of sugar



Licensing

Source of the graphics used I



Title: Hurricane Sandy Blackout New York Skyline
Author: David Shankbone
Source: https://commons.wikimedia.org/wiki/File:Hurricane_Sandy_Blackout_New_York_Skyline.JPG
License: CC BY 3.0

Title: Deepwater Horizon Offshore Drilling Platform on Fire
Author: ideum
Source: https://secure.flickr.com/photos/ideum/4711481781/
License: CC BY-SA 2.0

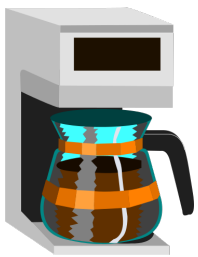
Title: DA-SC-88-01663
Author: imcomkorea
Source: https://secure.flickr.com/photos/imcomkorea/3017886760/
License: CC BY-NC-ND 2.0

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 of the graphics used II

Source: https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg
License: public domain



Title: Coffee machine drawing
Author: Ysangkok
Source: https://commons.wikimedia.org/wiki/File:Coffee_machine.svg
License: public domain

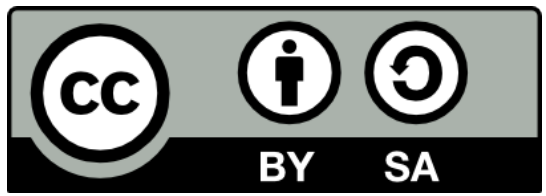
Title: taking a coffee break
Author: chris
Source: https://commons.wikimedia.org/wiki/File:SMirC-coffeebreak.svg
License: CC BY

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 on demand)

Author: **Étienne André**



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