Symbolic Monitoring against Specifications Parametric in Time and Data

Masaki Waga[1,2,3] , Étienne André[4,5,1], and Ichiro Hasuo[1,2]

National Institute of Informatics[1], SOKENDAI[2],
JSPS Research Fellow[3],
LIPN, Université Paris 13, CNRS[4], JFLI, UMI CNRS[5]

16 July 2019, CAV 2019

This work is partially supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603),
by JSPS Grants-in-Aid No. 15KT0012 & 18J22498 and by the ANR national research program PACS (ANR-14-CE28-0002).

Why Monitoring?

Exhaustive formal method

(e.g. model checking, reachability analysis)

- The system is correct/incorrect for **any** execution
- We need system model (**white box**)
- **Scalability** is a big issue

Monitoring

- The system is correct/incorrect for **the given** execution
- We do not need system model (**black box** is OK)
- Usually **scalable**

Spec. with Parameters

Concrete Spec. Example

the total amount of **7**-days withdrawal by user **Bob** should be < **1,000** USD

We do not know the best thresholds.
→ Parametrize and Synthesize!!
(Instead of True/False)

Parametric Spec. Example

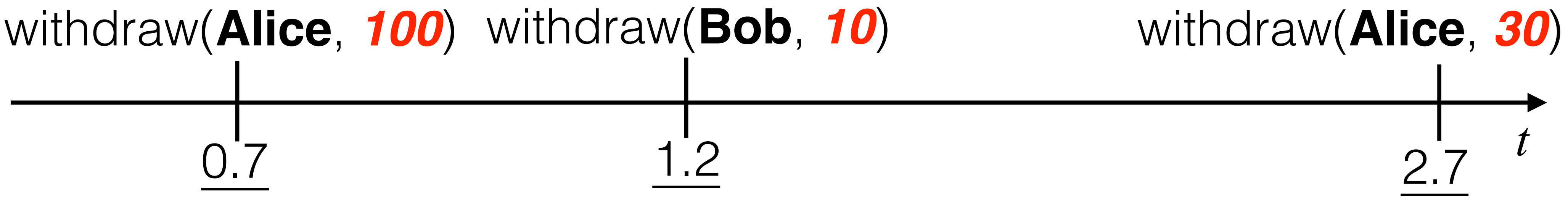
the total amount of p-days withdrawal by user N should be < T USD

Symbolic Monitoring

[Contribution]

Input

- **Time-series data**
  - System ***log*** (event + *data* + timestamp)
  - e.g.,


withdraw(**Alice**, ***100***)
withdraw(**Bob**, ***10***)
withdraw(**Alice**, ***30***)
0.7
1.2
2.7
t

- **Parameterized real-time spec. with data**
  - ***Spec.*** to be monitored
  - e.g., the total amount of $\boldsymbol{p}$-days withdrawal by user $\boldsymbol{N}$ should be < $\boldsymbol{T}$ USD

Output

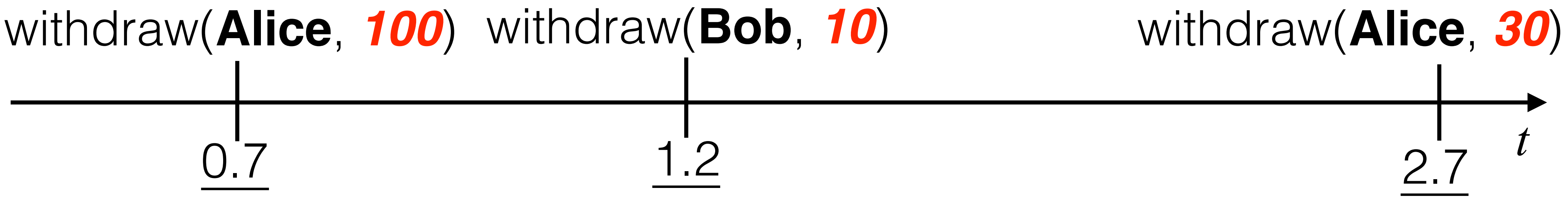
- **All** of the param. val. such that the ***log*** satisfies the ***spec.***
  - e.g., (N, T, p) = (**Alice**, ***140***, ***3.0***), (**Alice**, ***135***, ***4.0***), (**Bob**, ***20***, ***1.0***), …
  - **Infinitely** many → **Symbolic** representation

Symbolic Monitoring

[Contribution]

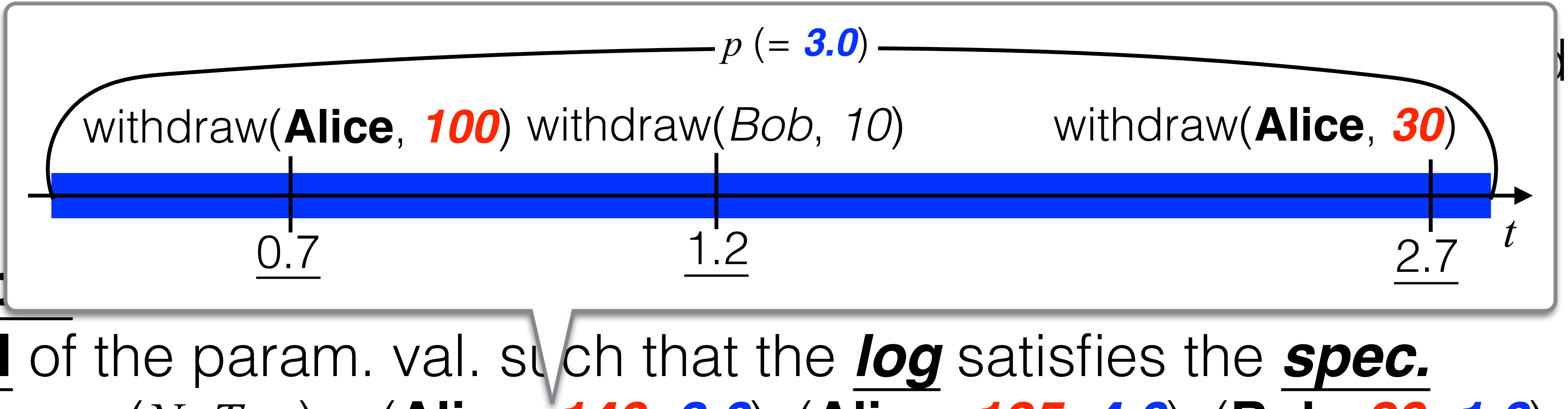
Input

- **Time-series data**
  - System ***log*** (event + *data* + timestamp)
  - e.g.,



withdraw(**Alice**, ***100***)
withdraw(**Bob**, ***10***)
withdraw(**Alice**, ***30***)
0.7
1.2
2.7
t

- **Parameterized real-time spec. with data**
  - ***Spec.*** to be monitored



p (= ***3.0***)
withdraw(**Alice**, ***100***) withdraw(*Bob*, *10*)
withdraw(**Alice**, ***30***)
0.7
1.2
2.7
t

Outp

- **All** of the param. val. such that the ***log*** satisfies the ***spec.***
  - e.g., (N, T, p) = (**Alice**, ***140***, ***3.0***), (**Alice**, ***135***, ***4.0***), (**Bob**, ***20***, ***1.0***), …
  - **Infinitely** many → **Symbolic** representation

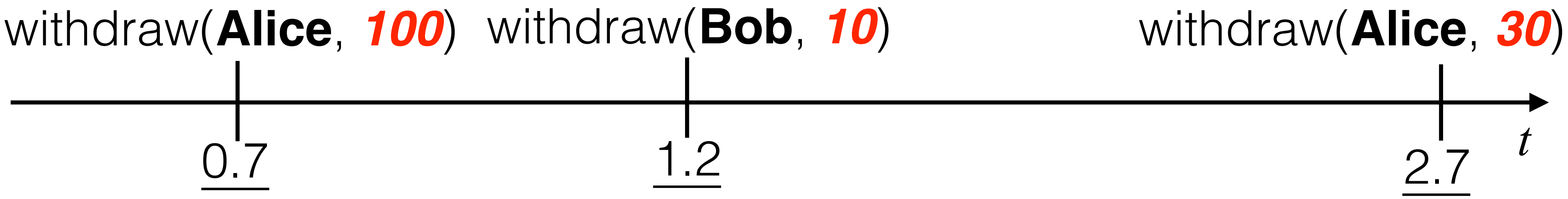
M. Waga (NII)

Symbolic Monitoring

[Contribution]

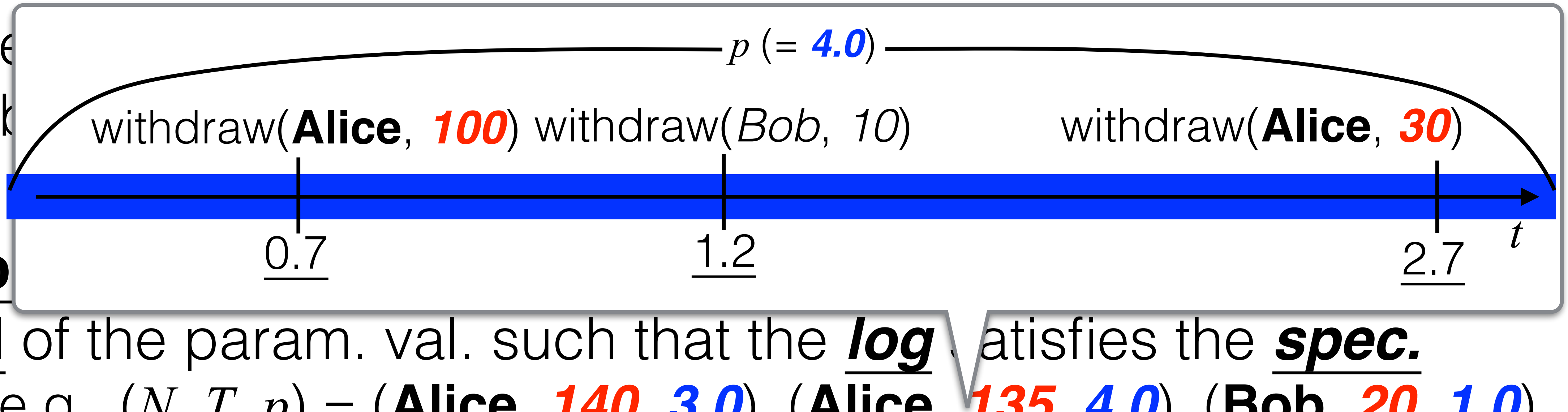
Input

- **Time-series data**
  - System ***log*** (event + *data* + timestamp)
  - e.g.,


withdraw(**Alice**, ***100***)
withdraw(**Bob**, ***10***)
withdraw(**Alice**, ***30***)
0.7
1.2
2.7
t

- **Parameterized real-time spec. with data**
  - ***Spec.*** to be monitored
  - e

b


p (= ***4.0***)
withdraw(**Alice**, ***100***) withdraw(*Bob*, *10*)
withdraw(**Alice**, ***30***)
0.7
1.2
2.7
t

Outp

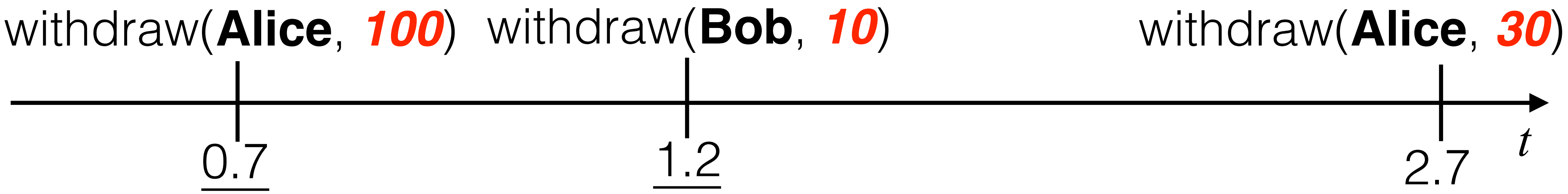
- **All** of the param. val. such that the ***log*** atisfies the ***spec.***
  - e.g., (N, T, p) = (**Alice**, ***140***, ***3.0***), (**Alice**, ***135***, ***4.0***), (**Bob**, ***20***, ***1.0***), …
  - **Infinitely** many → **Symbolic** representation

Symbolic Monitoring

[Contribution]

Input

- **Time-series data**
  - System ***log*** (event + *data* + timestamp)
  - e.g.,

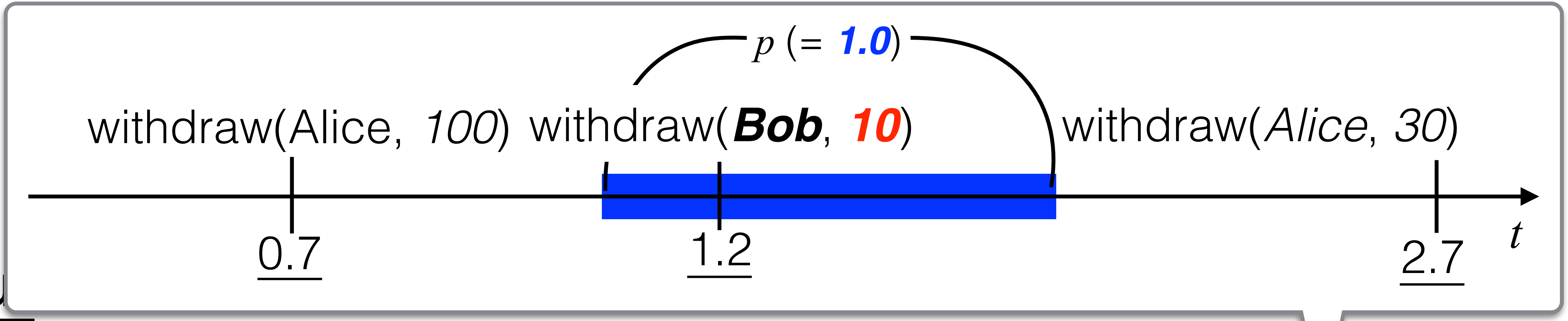


withdraw(**Alice**, ***100***) withdraw(**Bob**, ***10***) withdraw(**Alice**, ***30***)

0.7 1.2 2.7 t

- **Parameterized real-time spec. with data**
  - ***Spec.*** to be monitored
  - e

    b



p (= ***1.0***)

withdraw(Alice, *100*) withdraw(***Bob***, ***10***) withdraw(*Alice*, *30*)

0.7 1.2 2.7 t

Outpu

- **All** of the param. val. such that the ***log*** satisfies the ***spec.***
  - e.g., (N, T, p) = (**Alice**, ***140***, ***3.0***), (**Alice**, ***135***, ***4.0***), (**Bob**, ***20***, ***1.0***), …
  - **Infinitely** many → **Symbolic** representation

M. Waga (NII)

Contribution

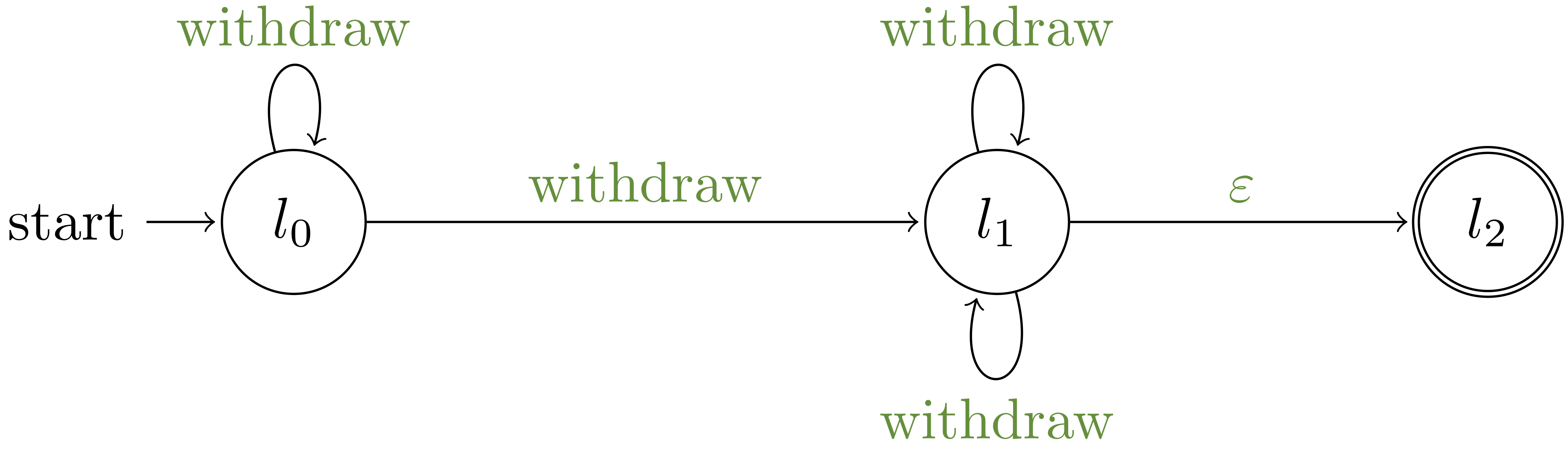
- Introduced **parametric timed data automata** (**PTDA**)
  - **PTDA**: Non-deterministic finite automata (NFA)
    + timing constraints + data + parameters
- Gave **symbolic monitoring** algorithm over a PTDA spec.
  - ***Symbolically*** synthesize **all** the feasible param. val. wrt. log
  - (Potentially) **infinitely** many param. val.
    → **symbolic** representation/operations
- Implementation + experiments → **Scalable**!!

Outline

- Motivation + Introduction
- Technical Part
  - Parametric timed data automata (PTDA)
    - PTDA: NFA + timing constraints + data + param.
  - Symbolic monitoring algorithm
    - Idea: follow trans. (+ non-deterministic branching)
- Experiments

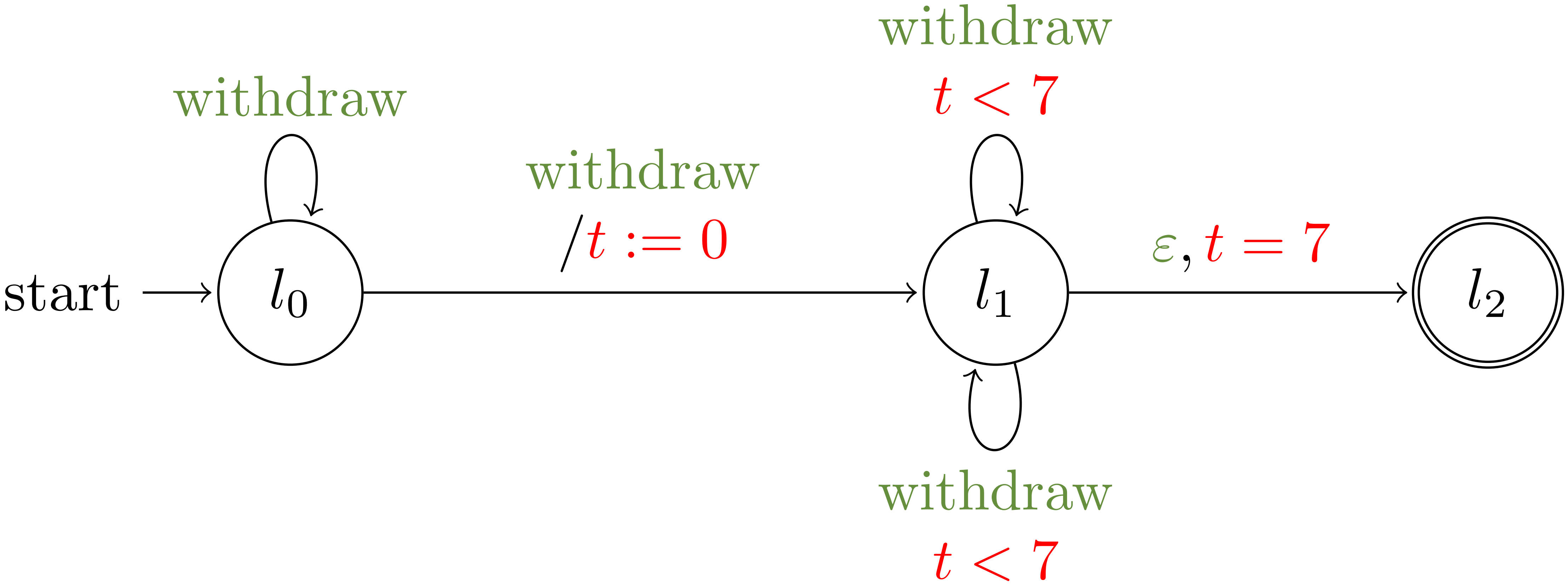
PTDA: **NFA** + time + data + parameters

NFA



PTDA: **NFA + time** + data + parameters

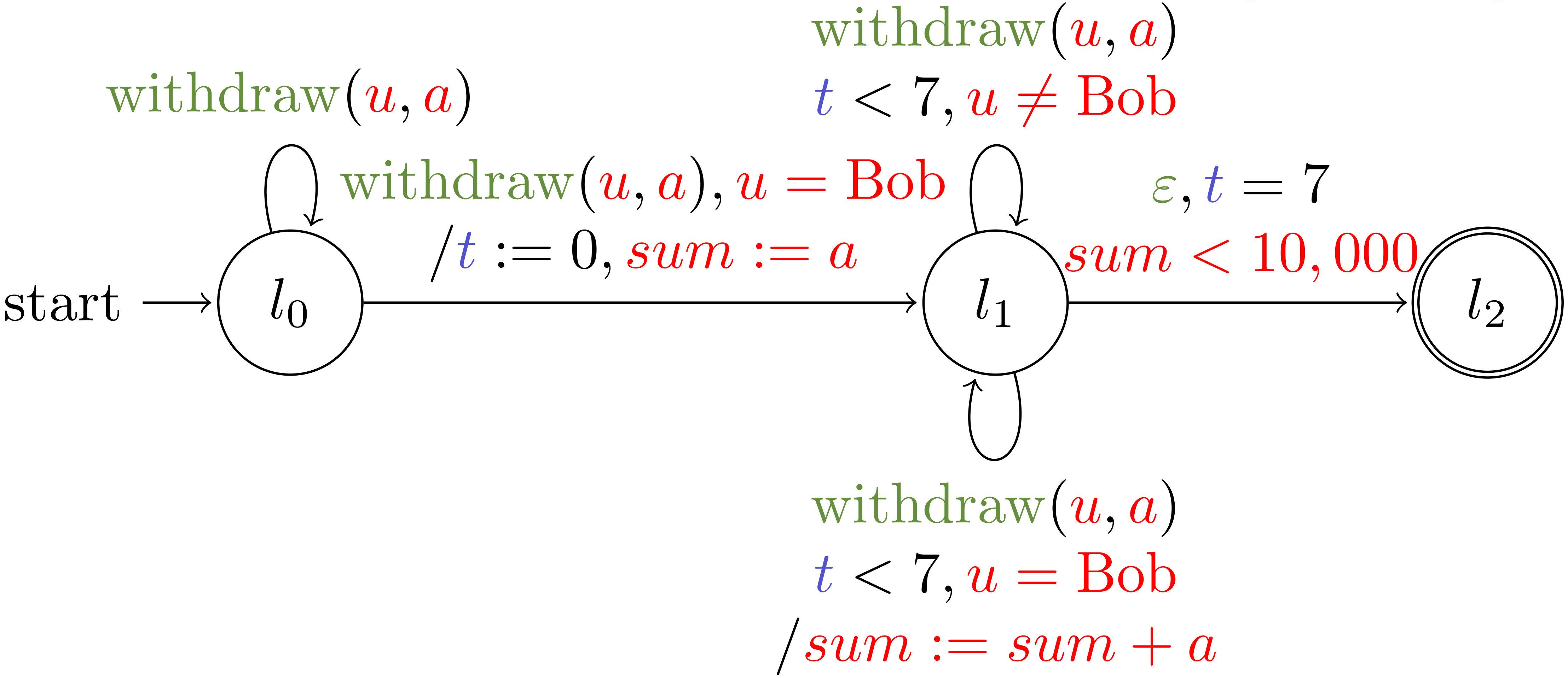
Timed Automaton (TA)



M. Waga (NII)

PTDA: **NFA + time + data** + parameters

Timed Data Automaton (TDA)

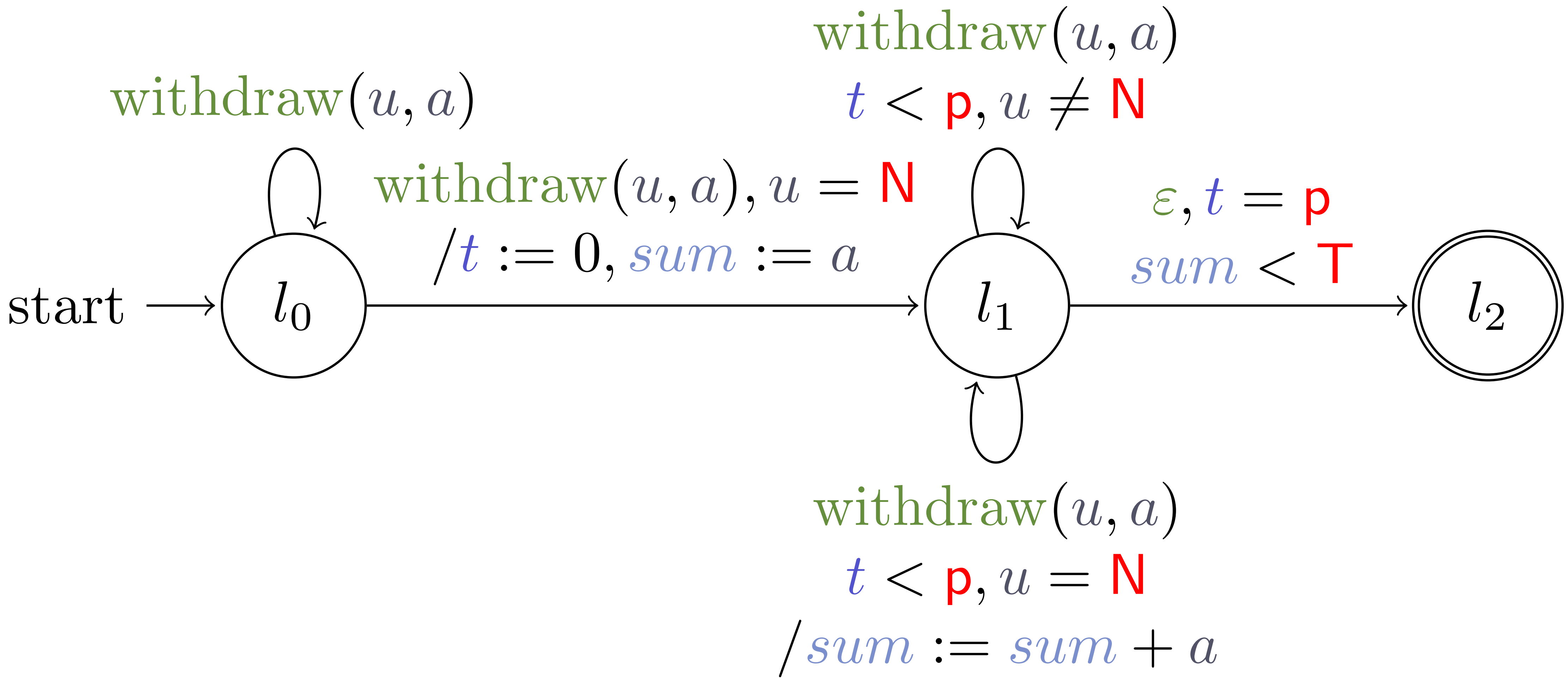


start
l_0
l_1
l_2
$\mathrm{withdraw}(u, a)$
$\mathrm{withdraw}(u, a), u = \mathrm{Bob}$
$/t := 0, sum := a$
$\mathrm{withdraw}(u, a)$
$t < 7, u \neq \mathrm{Bob}$
$\varepsilon, t = 7$
$sum < 10,000$
$\mathrm{withdraw}(u, a)$
$t < 7, u = \mathrm{Bob}$
$/sum := sum + a$

M. Waga (NII)

PTDA: **NFA + time + data + parameters**

Parametric Timed Data Automaton (PTDA)



Data Type ($\mathbb{D}$, $\mathcal{DE}$, $\mathcal{DU}$)

$\mathbb{D}$: infinite domain

$\mathcal{DE}$: Boolean expression (for guards)

$\mathcal{DU}$: updates (for variable updates/assignments)

- (Explained Later) Our symbolic monitoring algorithm works for any data type with some symbolic operations
    - e.g., Strings ($\mathbb{S}$), Rationals ($\mathbb{Q}$), …

Outline

- Motivation + Introduction
- Technical Part
  - Parametric timed data automata (PTDA)
    - PTDA: NFA + timing constraints + data + param.
  - Symbolic monitoring algorithm
    - Idea: follow trans. (+ non-deterministic branching)
- Experiments

Idea of our Symbolic Monitoring Algorithm

follow the transitions of PTDA

+

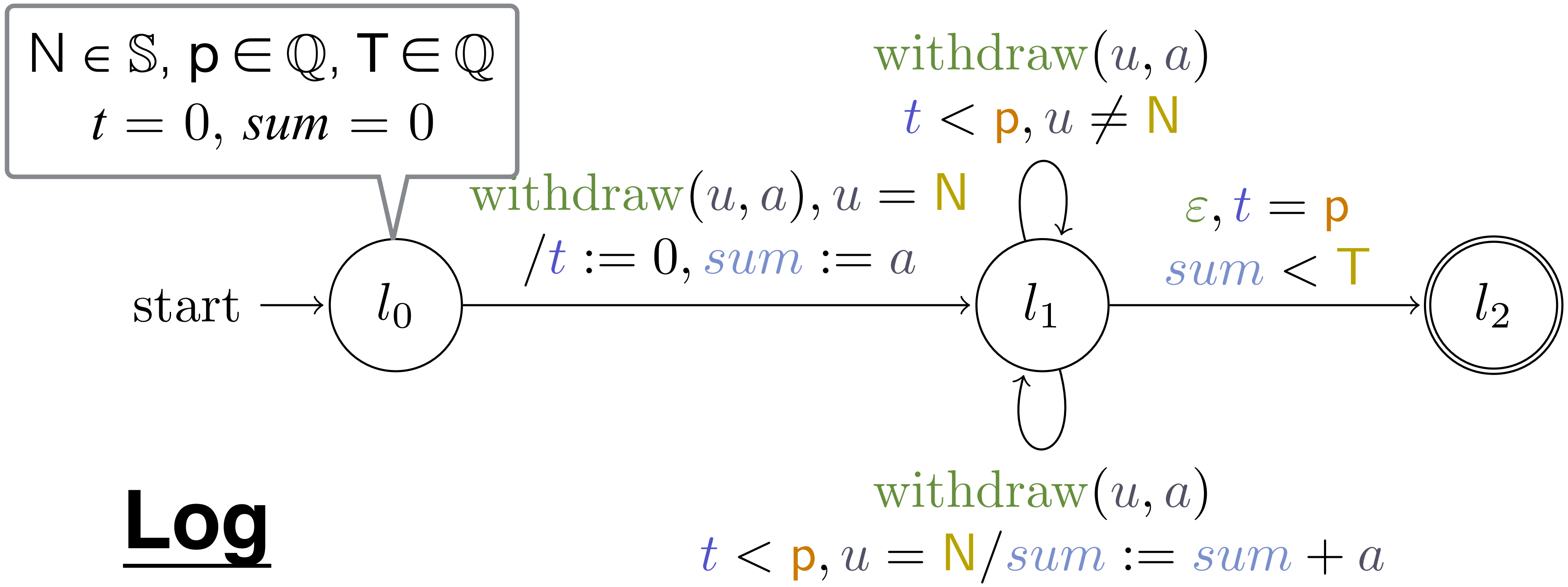
abstraction of clock/data/param. val.

(e.g., by convex polyhedra or lists of forbidden strings)

+

(Non-deterministic branching by breadth first search)

Symbolic Monitoring by Following Transitions



$\mathsf{N} \in \mathbb{S}, \mathsf{p} \in \mathbb{Q}, \mathsf{T} \in \mathbb{Q}$
$t = 0, \mathit{sum} = 0$
start
l_0
$\mathrm{withdraw}(u, a), u = \mathsf{N}$
$/ t := 0, \mathit{sum} := a$
l_1
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u \neq \mathsf{N}$
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u = \mathsf{N} / \mathit{sum} := \mathit{sum} + a$
$\varepsilon, t = \mathsf{p}$
$\mathit{sum} < \mathsf{T}$
l_2

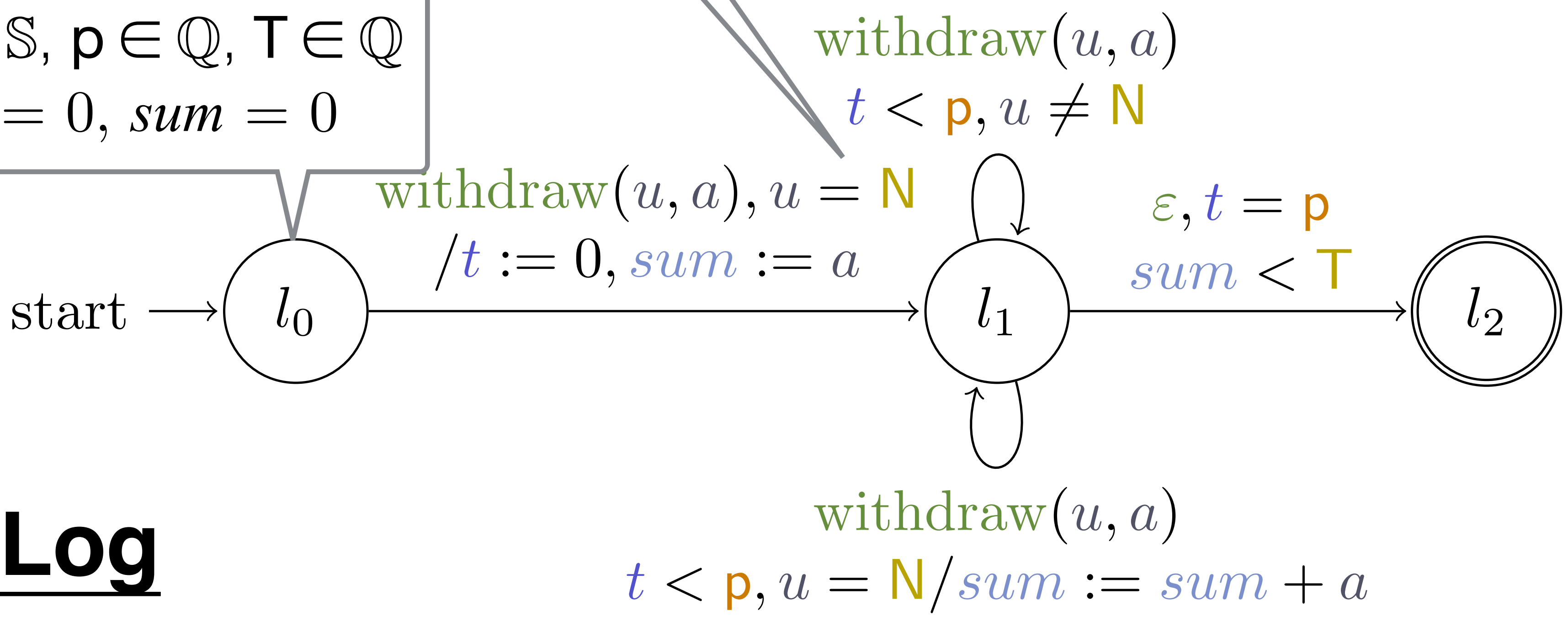
Log

withdraw(*Alice*, *100*) @0.7
withdraw(*Bob*, *10*) @1.2
withdraw(*Alice*, *30*) @2.7

Symbolic Monitoring by Following Transitions

N = Alice, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0,\ \mathit{sum} = 100$

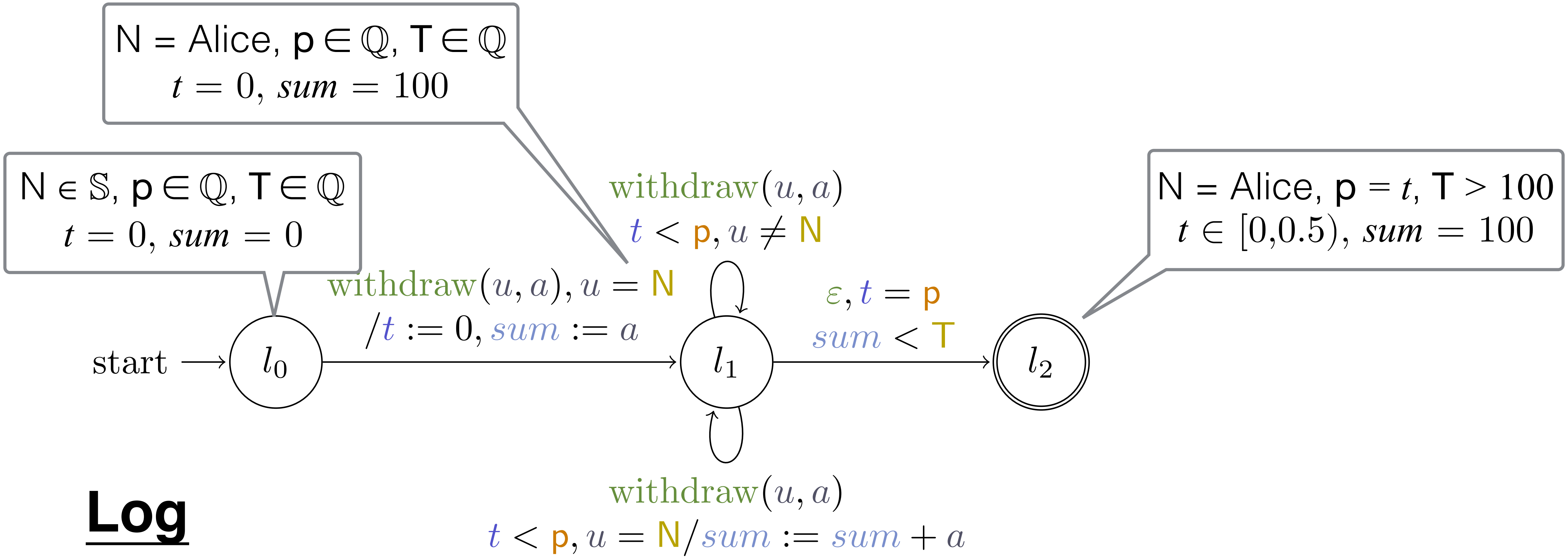
N $\in \mathbb{S}$, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0,\ \mathit{sum} = 0$


$\mathrm{withdraw}(u, a)$
$t <$ p$, u \neq$ N
$\mathrm{withdraw}(u, a), u =$ N
$/t := 0, \mathit{sum} := a$
$\varepsilon, t =$ p
$\mathit{sum} <$ T
start
l_0
l_1
l_2
$\mathrm{withdraw}(u, a)$
$t <$ p$, u =$ N$/\mathit{sum} := \mathit{sum} + a$

Log

```
→withdraw(Alice, 100) @0.7
  withdraw(Bob, 10) @1.2
  withdraw(Alice, 30) @2.7
```

Symbolic Monitoring by Following Transitions



N = Alice, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 100$

N $\in \mathbb{S}$, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 0$

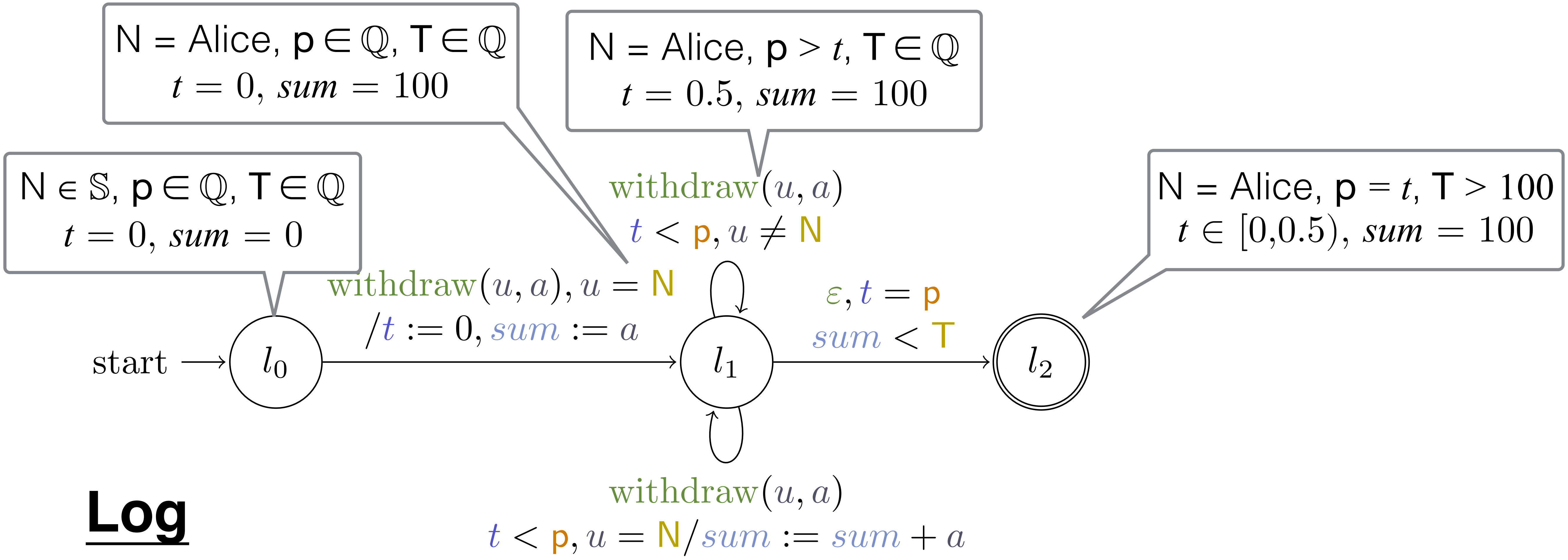
N = Alice, p $= t$, T > 100
$t \in [0,0.5)$, $sum = 100$

start
l_0
$\mathrm{withdraw}(u, a), u = \mathsf{N}$
$/ t := 0, sum := a$
l_1
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u \neq \mathsf{N}$
$\varepsilon, t = \mathsf{p}$
$sum < \mathsf{T}$
l_2
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u = \mathsf{N} / sum := sum + a$

Log

→withdraw(*Alice*, *100*) @0.7
withdraw(*Bob*, *10*) @1.2
withdraw(*Alice*, *30*) @2.7

Symbolic Monitoring by Following Transitions



N $\in \mathbb{S}$, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 0$

N = Alice, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 100$

N = Alice, p $> t$, T $\in \mathbb{Q}$
$t = 0.5$, $sum = 100$

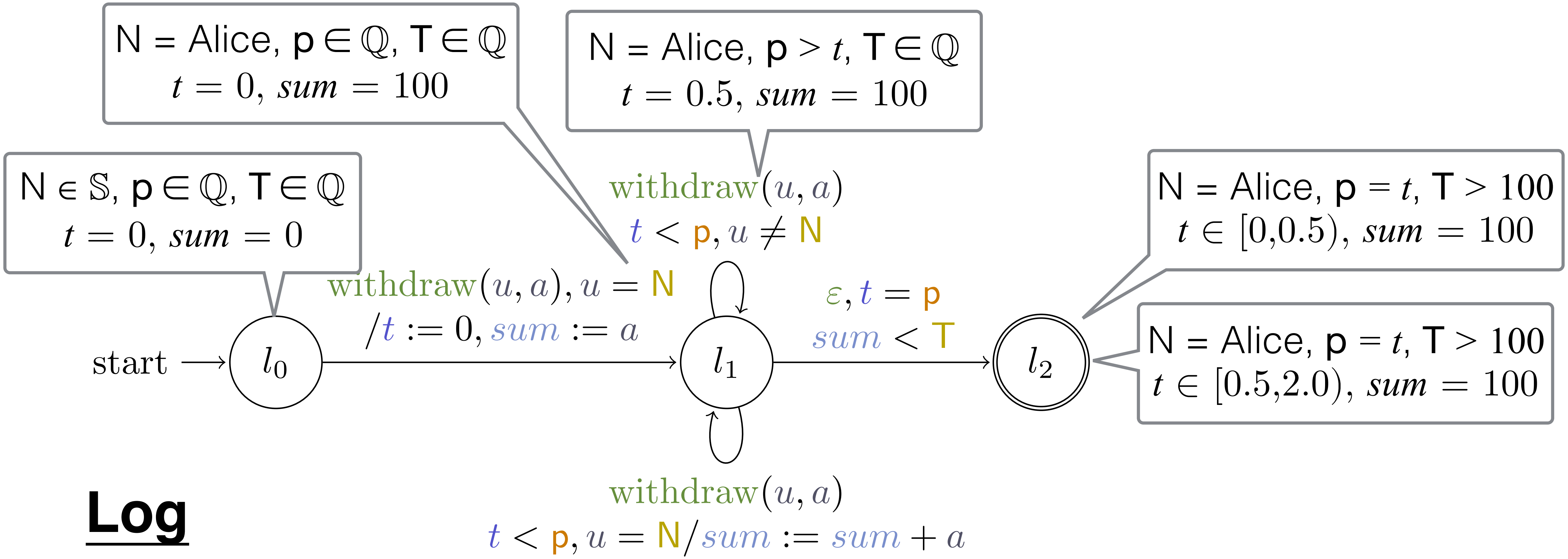
N = Alice, p $= t$, T > 100
$t \in [0,0.5)$, $sum = 100$

start
l_0
l_1
l_2
$\mathrm{withdraw}(u, a), u = \mathsf{N}$
$/t := 0, sum := a$
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u \neq \mathsf{N}$
$\varepsilon, t = \mathsf{p}$
$sum < \mathsf{T}$
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u = \mathsf{N}/sum := sum + a$

Log

withdraw(*Alice*, *100*) @0.7
→withdraw(*Bob*, *10*) @1.2
withdraw(*Alice*, *30*) @2.7

Symbolic Monitoring by Following Transitions



start
l_0
l_1
l_2
$\mathrm{withdraw}(u,a), u = \mathsf{N}$
$/t := 0, sum := a$
$\mathrm{withdraw}(u,a)$
$t < \mathsf{p}, u \neq \mathsf{N}$
$\varepsilon, t = \mathsf{p}$
$sum < \mathsf{T}$
$\mathrm{withdraw}(u,a)$
$t < \mathsf{p}, u = \mathsf{N} / sum := sum + a$

N ∊ $\mathbb{S}$, p ∈ $\mathbb{Q}$, T ∈ $\mathbb{Q}$
$t = 0$, $sum = 0$

N = Alice, p ∈ $\mathbb{Q}$, T ∈ $\mathbb{Q}$
$t = 0$, $sum = 100$

N = Alice, p $> t$, T ∈ $\mathbb{Q}$
$t = 0.5$, $sum = 100$

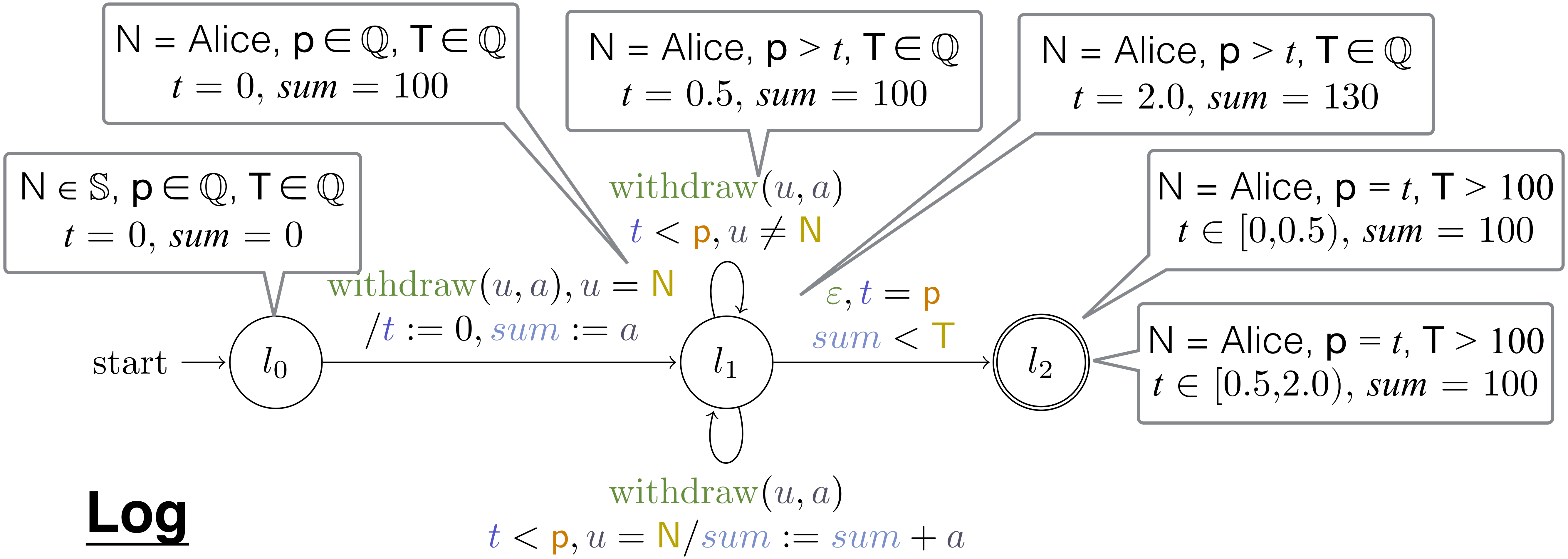
N = Alice, p $= t$, T > 100
$t \in [0{,}0.5)$, $sum = 100$

N = Alice, p $= t$, T > 100
$t \in [0.5{,}2.0)$, $sum = 100$

Log

withdraw(*Alice*, *100*) @0.7
→withdraw(*Bob*, *10*) @1.2
withdraw(*Alice*, *30*) @2.7

Symbolic Monitoring by Following Transitions



N $\in \mathbb{S}$, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 0$

N = Alice, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 100$

N = Alice, p $> t$, T $\in \mathbb{Q}$
$t = 0.5$, $sum = 100$

N = Alice, p $> t$, T $\in \mathbb{Q}$
$t = 2.0$, $sum = 130$

N = Alice, p $= t$, T > 100
$t \in [0,0.5)$, $sum = 100$

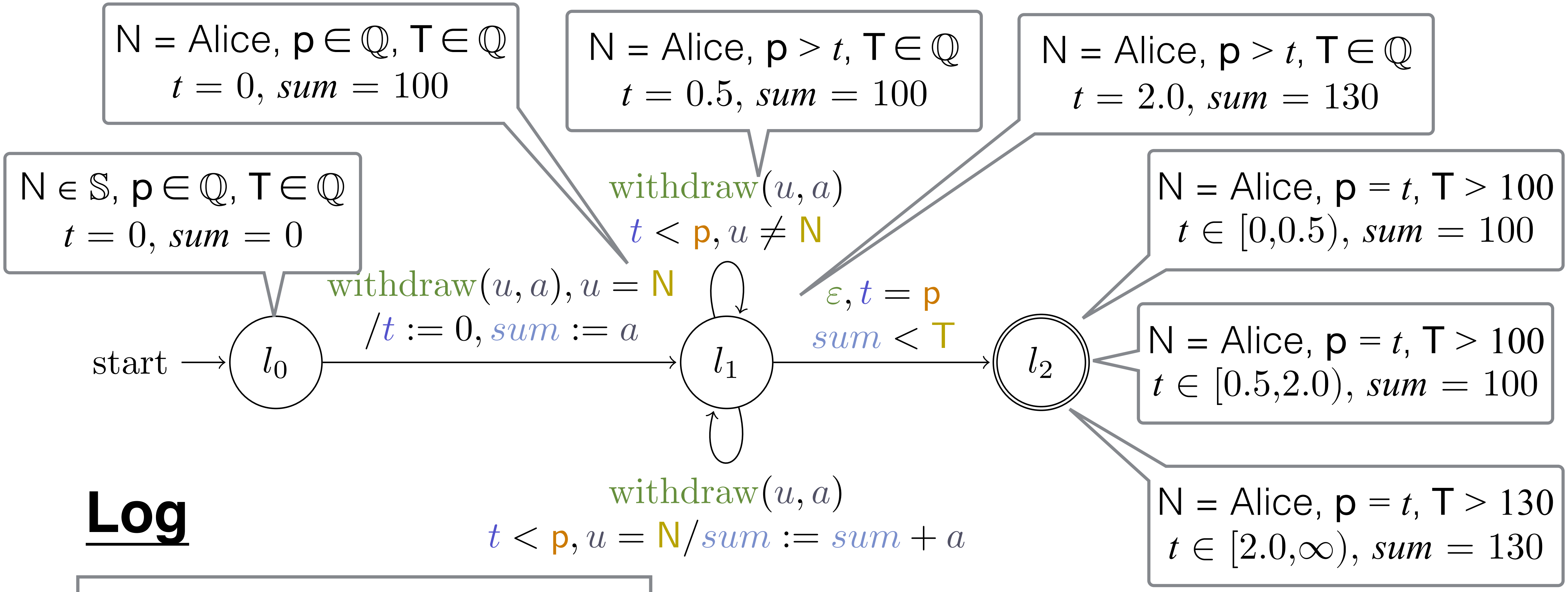
N = Alice, p $= t$, T > 100
$t \in [0.5,2.0)$, $sum = 100$

start
l_0
l_1
l_2
$\mathrm{withdraw}(u, a), u = \mathsf{N}$
$/t := 0, sum := a$
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u \neq \mathsf{N}$
$\varepsilon, t = \mathsf{p}$
$sum < \mathsf{T}$
$\mathrm{withdraw}(u, a)$
$t < \mathsf{p}, u = \mathsf{N} / sum := sum + a$

Log

```
  withdraw(Alice, 100) @0.7
  withdraw(Bob, 10) @1.2
→withdraw(Alice, 30) @2.7
```

Symbolic Monitoring by Following Transitions



N ∈ $\mathbb{S}$, p ∈ $\mathbb{Q}$, T ∈ $\mathbb{Q}$
$t = 0$, $sum = 0$

N = Alice, p ∈ $\mathbb{Q}$, T ∈ $\mathbb{Q}$
$t = 0$, $sum = 100$

N = Alice, p $> t$, T ∈ $\mathbb{Q}$
$t = 0.5$, $sum = 100$

N = Alice, p $> t$, T ∈ $\mathbb{Q}$
$t = 2.0$, $sum = 130$

N = Alice, p $= t$, T > 100
$t \in [0,0.5)$, $sum = 100$

N = Alice, p $= t$, T > 100
$t \in [0.5,2.0)$, $sum = 100$

N = Alice, p $= t$, T > 130
$t \in [2.0,\infty)$, $sum = 130$

start $\longrightarrow$ l_0

$\mathrm{withdraw}(u, a), u =$ N
$/t := 0, sum := a$

l_1

$\mathrm{withdraw}(u, a)$
$t <$ p, $u \neq$ N

$\mathrm{withdraw}(u, a)$
$t <$ p, $u =$ N$/sum := sum + a$

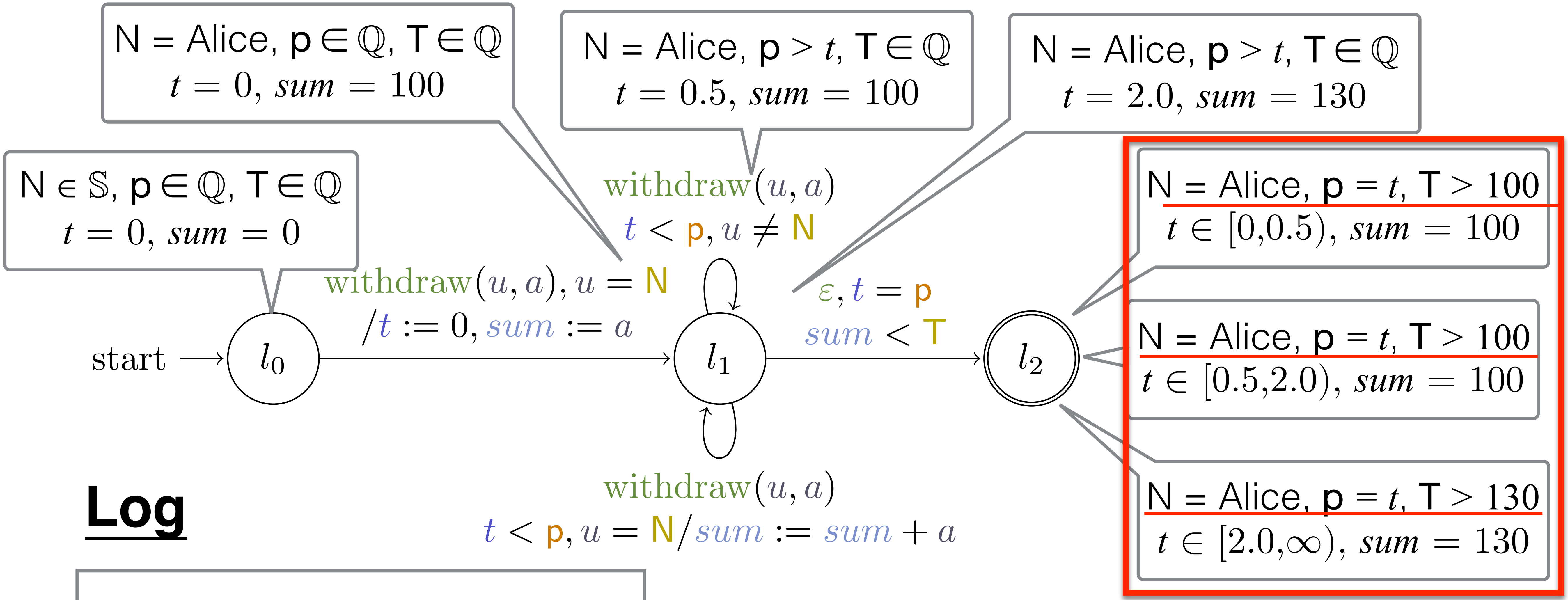
$\varepsilon, t =$ p
$sum <$ T

l_2

Log

```
   withdraw(Alice, 100) @0.7
   withdraw(Bob, 10) @1.2
→withdraw(Alice, 30) @2.7
```

Symbolic Monitoring by Following Transitions



start
l_0
l_1
l_2
$\mathrm{withdraw}(u,a), u = \mathsf{N}$
$/t := 0, sum := a$
$\mathrm{withdraw}(u,a)$
$t < \mathsf{p}, u \neq \mathsf{N}$
$\mathrm{withdraw}(u,a)$
$t < \mathsf{p}, u = \mathsf{N} / sum := sum + a$
$\varepsilon, t = \mathsf{p}$
$sum < \mathsf{T}$

N $\in \mathbb{S}$, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 0$

N = Alice, p $\in \mathbb{Q}$, T $\in \mathbb{Q}$
$t = 0$, $sum = 100$

N = Alice, p $> t$, T $\in \mathbb{Q}$
$t = 0.5$, $sum = 100$

N = Alice, p $> t$, T $\in \mathbb{Q}$
$t = 2.0$, $sum = 130$

N = Alice, p $= t$, T > 100
$t \in [0,0.5)$, $sum = 100$

N = Alice, p $= t$, T > 100
$t \in [0.5,2.0)$, $sum = 100$

N = Alice, p $= t$, T > 130
$t \in [2.0,\infty)$, $sum = 130$

Log

withdraw(*Alice*, *100*) @0.7
withdraw(*Bob*, *10*) @1.2
→withdraw(*Alice*, *30*) @2.7

Result

M. Waga (NII)

Termination of Symbolic Monitoring

Thm.
Our symbolic monitoring algorithm terminates for any data types ($\mathbb{D}$, $\mathcal{DE}$, $\mathcal{DU}$) such that we can compute

restriction, update, emptiness checking, and projection.

Examples

- Strings ($\mathbb{S}$) with lists (of forbidden strings)
- Rationals ($\mathbb{Q}$) with convex polyhedra

Outline

- Motivation + Introduction
- Technical Part
  - Parametric timed data automata (PTDA)
    - PTDA: NFA + timing constraints + data + param.
  - Symbolic monitoring algorithm
    - Idea: follow trans. (+ non-deterministic branching)
- Experiments

Environment of Experiments

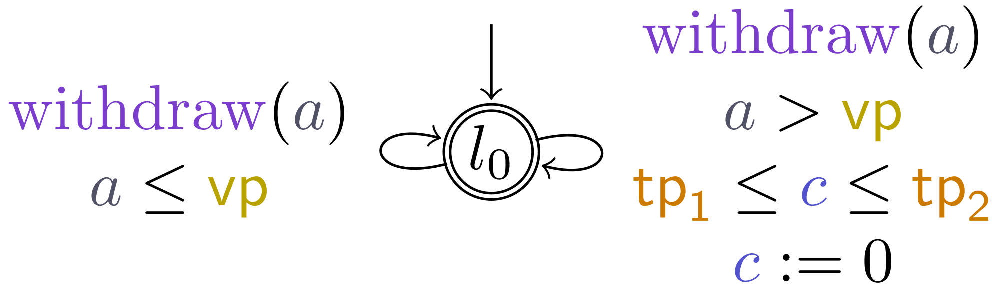
- **Data**: strings (by lists) and rationals (by convex polyhedra)
- Used 3 original benchmarks:
  - **Copy**: "The value of a signal should be copied to another signal"
    - Inspired by [Brim+, Information and Computation 236]
  - **Dominant**: "Detect a dominant withdrawal by a user"
    - Inspired by [Basin+, RV-CuBES'17]
  - **Periodic**: Synthesize periods of periodic withdrawals (Explained later)
- Amazon EC2 c4.large instance / Ubuntu 18.04 LTS (64 bit)
  - 2.9 GHz Intel Xeon E5-2666 v3, 2 vCPUs, 3.75 GiB RAM

“Periodic" Benchmark

Input 1: Log

- small withdrawals occurs every 5±1 time units
- middle withdrawals occurs every 50±3 time units
- large withdrawals occurs every 100±5 time units

Input 2: Spec.



$\mathrm{withdraw}(a)$

$a \leq \mathsf{vp}$

l_0

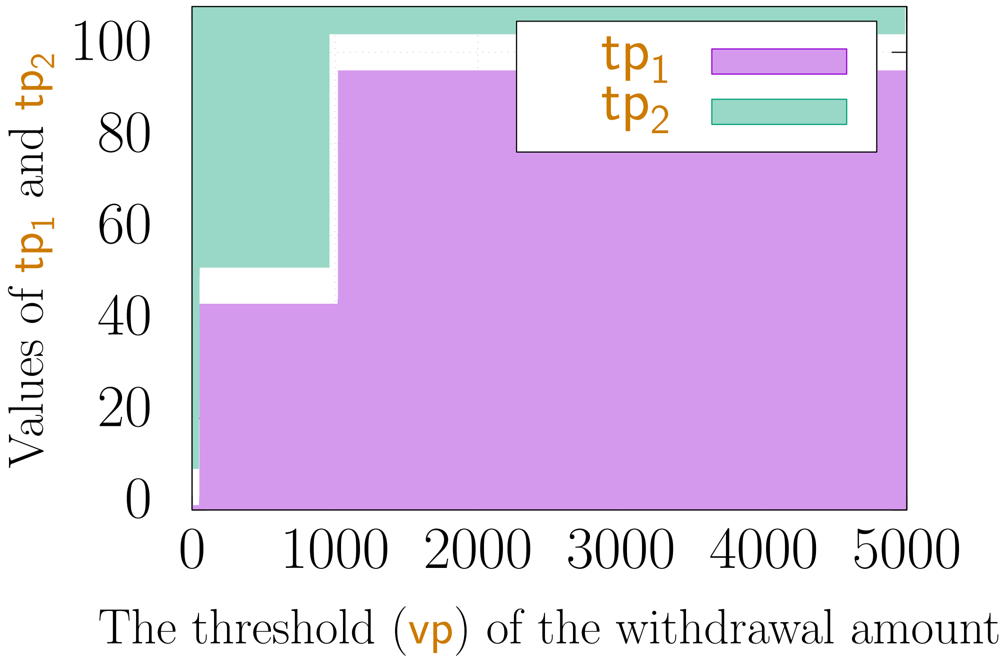
$\mathrm{withdraw}(a)$

$a > \mathsf{vp}$

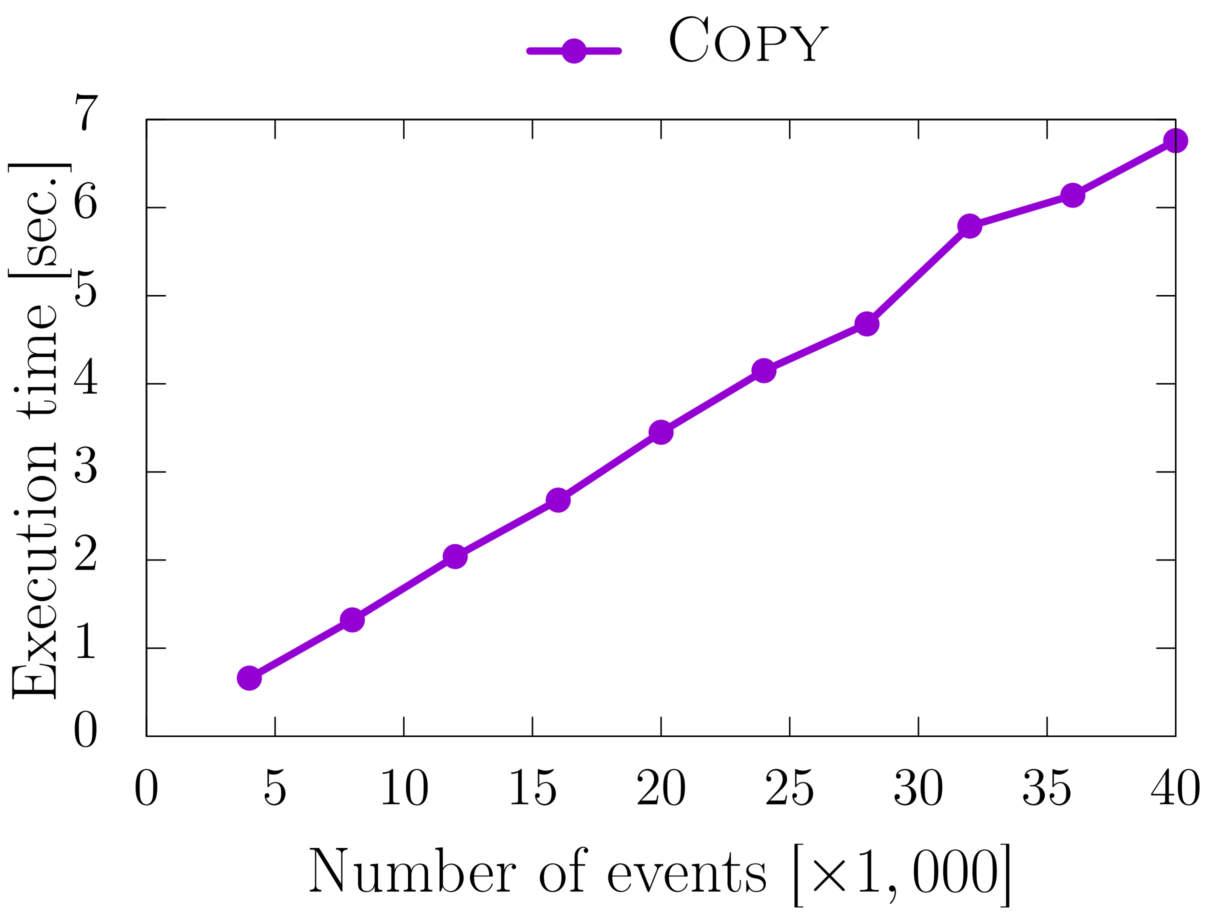
$\mathsf{tp}_1 \leq c \leq \mathsf{tp}_2$

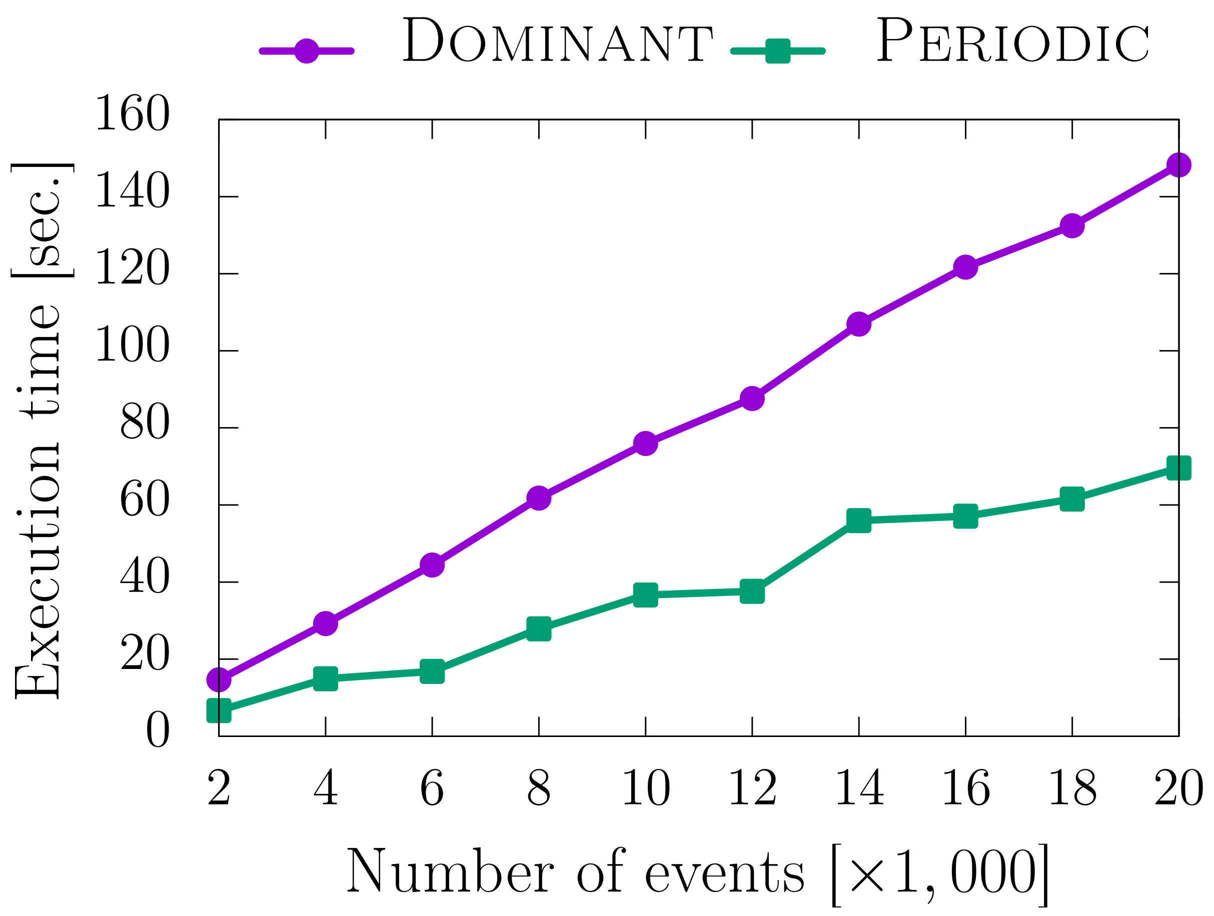
$c := 0$

Result



Execution Time





- 20,000 entries in 1 - 2 min.
- Execution time is **linear** in all of three benchmarks
  - Much more efficient than the worst case!!

M. Waga (NII)

Related Works

- Symbolic Register Automata [D'Antoni+, CAV 2019]
  - **Register** to remember previous information
- MFOTL [Basin+, J. ACM 62(2) 2015] [Basin+, RV-CuBES 2017] (MonPoly)
  - Many common concepts
    - timing constraints, quantified variables, and aggregation
  - **Concrete** outputs
- PSTL [Asarin+, RV 2011], [Bakhirkin+, HSCC 2018]
  - It **synthesizes** the parameter valuations
  - Specific to **real**-values / Signal-based
- QTL [Havelund+, FMCAD 2017], [Havelund+, MT-CPS'18] (DejaVu)
  - They use **BDD** in monitoring, though their outputs are rather **concrete**
  - no native support of timestamps

Conclusion

- Introduced parametric timed data automata **(PTDA)**
  - **PTDA:** NFA + timing constraints + data + parameters
- Gave symbolic monitoring algorithm over PTDA
  - **Idea**: follow trans. (+ non-deterministic branching)
- Implementation + experiments
  - about 20,000 entries in 1 or 2 min

Future Works

- Use BDD for more “symbolic” monitoring
- Employ polarity for further efficiency
    - Polarity: Either only expr < p or only expr > p
- Inference when $\mathbb{D}$ is finite
    - If $\mathbb{D} = \{a,b,c\}$, neither a nor b ⇒ c