

# Statistical Model Checking in UPPAAL

Alexandre David, Kim G. Larsen,  
Axel Legay, Marius Mikucionis  
Wang Zheng, Peter Bulychev,  
Jonas van Vliet, Danny Poulsen,  
Dehui Du, Guangyuan Li



CAV 11, PDMC 11, FORMATS 11,  
QAPL12, LPAR12, iWIGPL12, RV12,  
FORMATS12, HBS12, ISOLA12,  
SCIENCE China, NFM13, RV13, AVOCS13



# UPPAAL

Safety ✓

$A[] \text{ forall } (i : \text{id\_t}) \text{ forall } (j : \text{id\_t})$   
 $\text{Train}(i).\text{Cross} \ \&\& \ \text{Train}(j).\text{Cross} \ \text{imply } i == j$

Reachability ✓

$E \leftrightarrow \text{Train}(0).\text{Cross} \ \text{and} \ \text{Train}(1).\text{Stop}$

Liveness ✓

$\text{Train}(0).\text{Appr} \ \text{-->} \ \text{Train}(0).\text{Cross}$

Limited quantitative analysis ✓

$A \leftrightarrow .. \ E[] ..$  ✓

sup: .. inf: ..

Performance properties ✗

$\text{Pr}[ \leftrightarrow \text{Time} \leq 500 \ \text{and} \ \text{Train}(0).\text{Cross} ] \geq 0.7$   
 $\text{Pr}[ \text{Train}(0).\text{Appr} \ \text{-->}_{\text{Time} \leq 100} \ \text{Train}(0).\text{Cross} ] \geq 0.4$

State-space explosion ✗

# UPPAAL SMC

Performance properties ✓

$$\Pr[ \leq 200 ]( \langle \rangle \text{Train}(5).\text{Cross} )$$

$$\Pr[ \leq 100 ]( \langle \rangle \text{Train}(0).\text{Cross} ) \geq 0.8$$

$$\Pr[ \leq 100 ]( \langle \rangle \text{Train}(5).\text{Cross} ) \geq$$

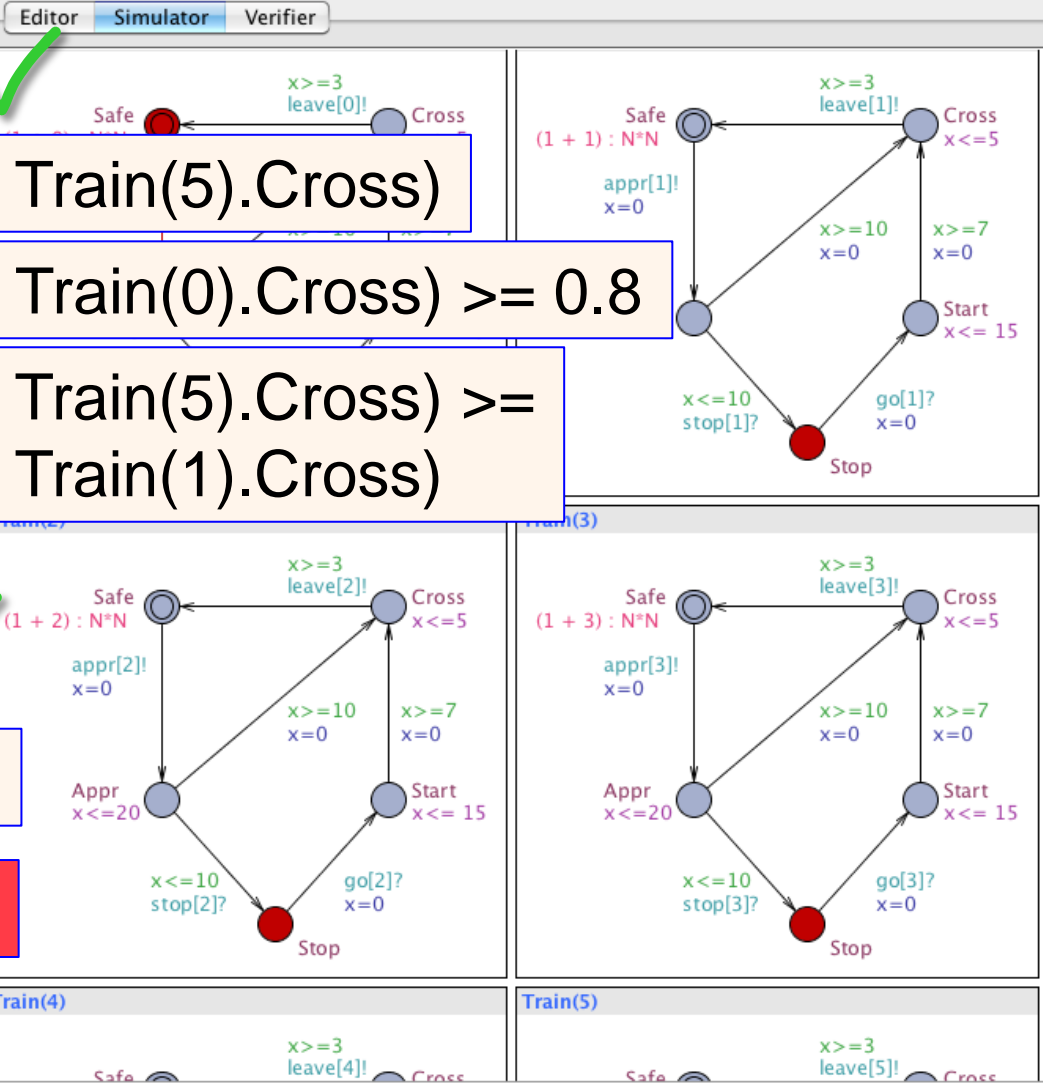
$$\Pr[ \leq 100 ]( \langle \rangle \text{Train}(1).\text{Cross} )$$

State-space explosion ✓

Generate runs

Performance properties

State-space explosion



# Overview

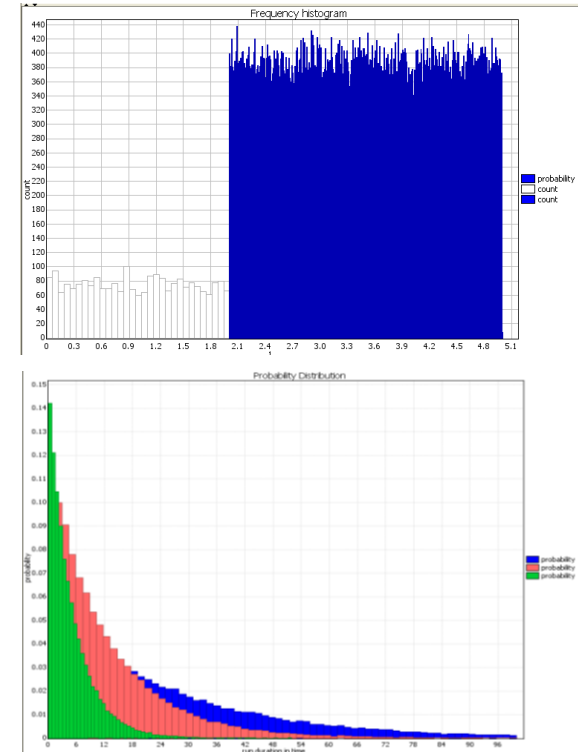
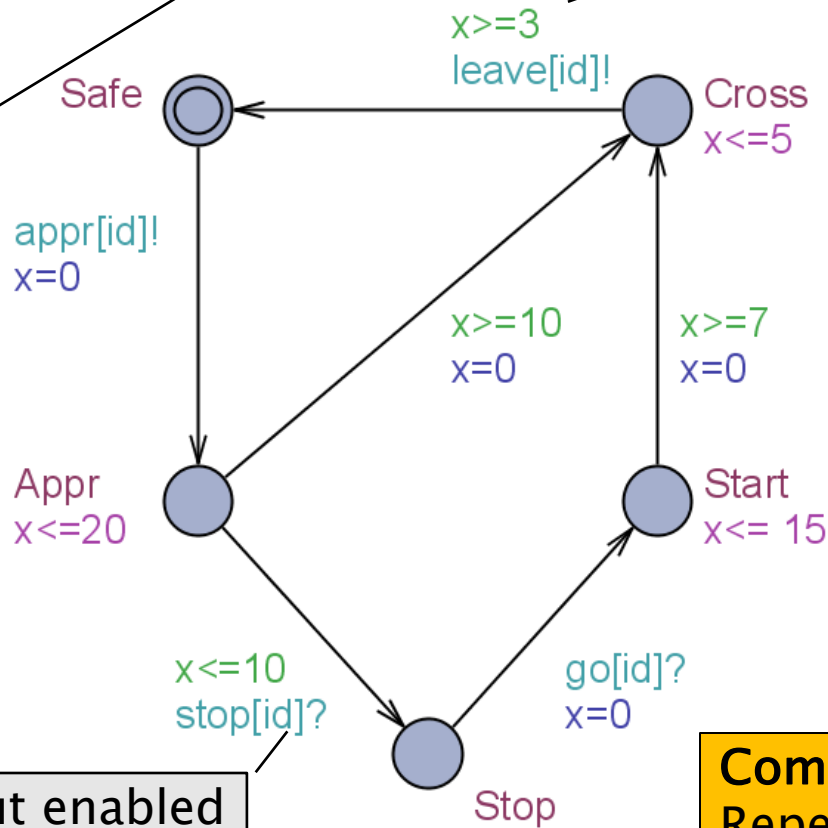
- Stochastic Semantics of Networks of Timed Automata
- Statistical Model Checking in UPPAAL
  - Estimation
  - Sequential Hypothesis Testing
  - Sequential Probability Comparison
  - Parameterized Probability Comparison
- SMC of Hybrid Automata
- Case Studies & Demo



# Stochastic Semantics of TA

Exponential Distribution

Uniform Distribution

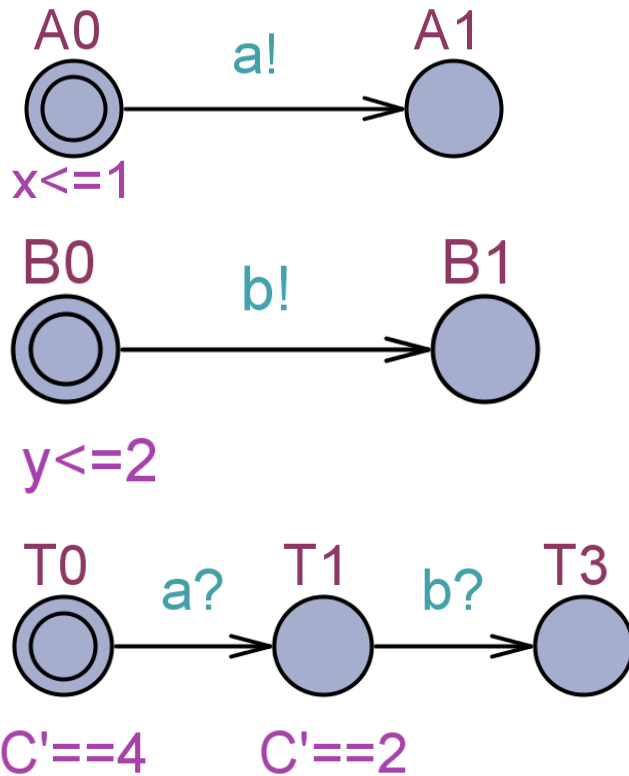


Input enabled

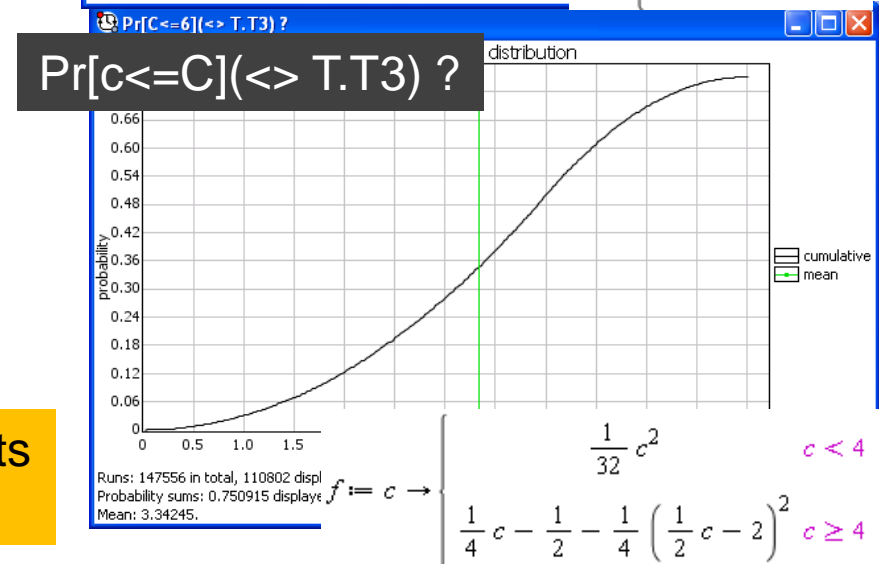
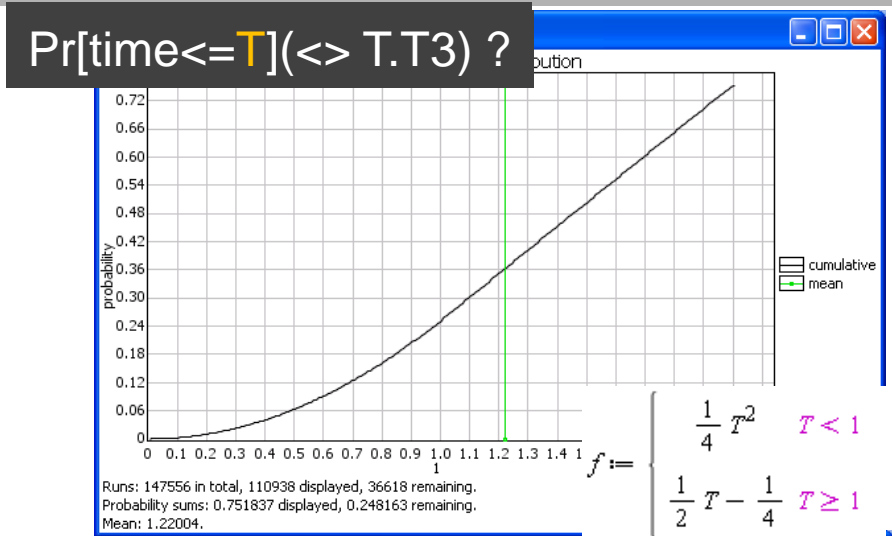
**Composition =**  
Repeated races between components



# Stochastic Semantics of Timed Automata

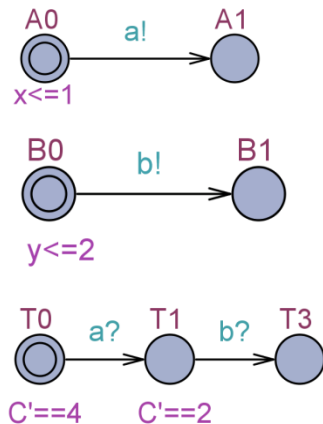


**Composition** = Race between components for outputting



# Stochastic Semantics of Timed Automata

$\mathcal{A}$



## Assumptions:

Component TAs are:

- Input enabled
- Deterministic
- Disjoint set of output actions

$\pi(\mathbf{s}, a_1 a_2 \dots a_n)$ :

the set of maximal runs from  $\mathbf{s}$  with a prefix

$t_1 a_1 t_2 a_2 \dots t_n a_k$

for some  $t_1, \dots, t_n \in \mathbf{R}$ .

$\mathbb{P}_{\mathcal{A}}(\pi(\mathbf{s}, a_1 a_2 \dots a_n)) =$

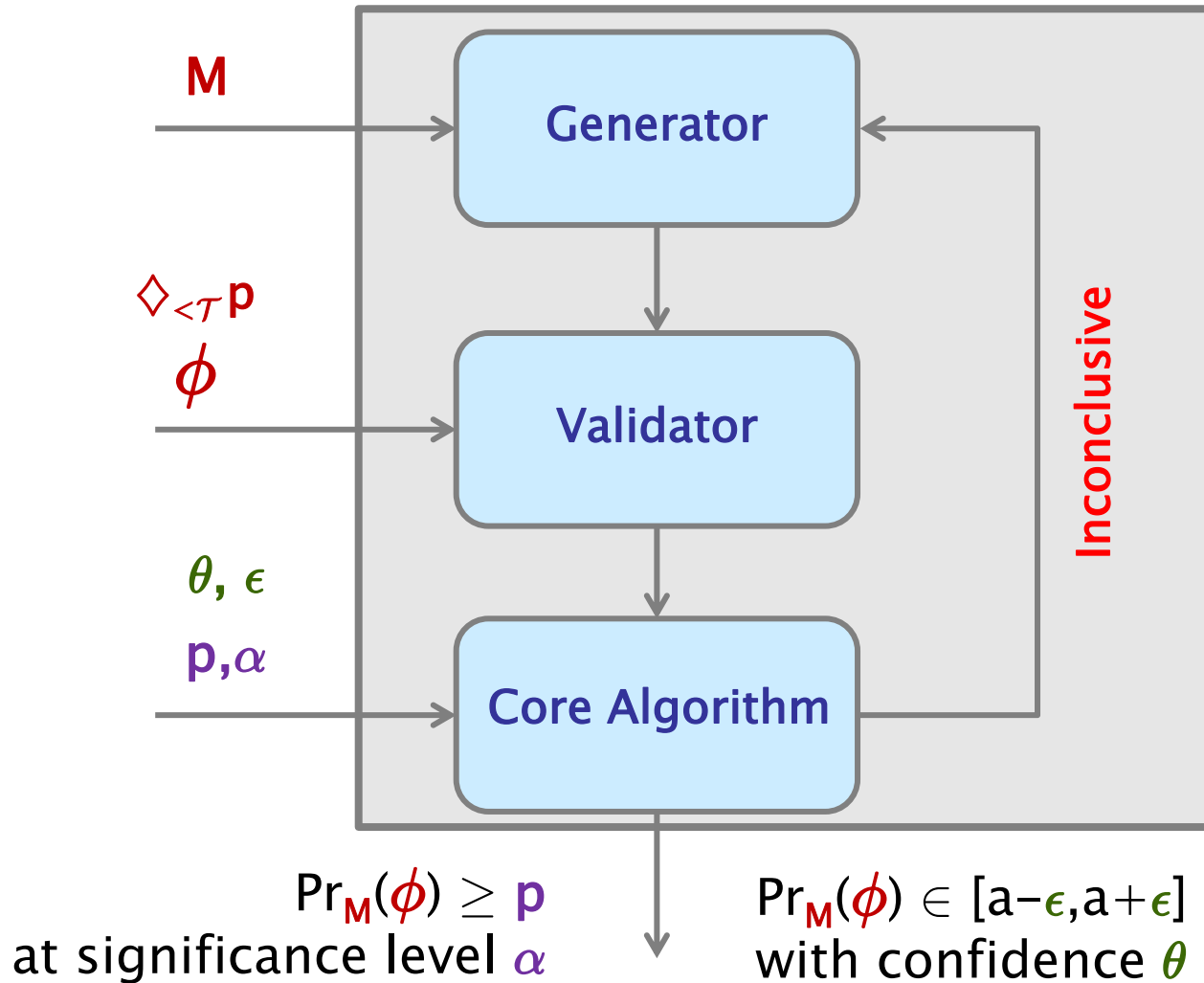
$$\int_{t \geq 0} \mu_{s_c}(t) \cdot \left( \prod_{j \neq c} \int_{\tau > t} \mu_{s_j}(\tau) d\tau \right) \cdot \gamma_{s_c t}(a_1) \cdot \mathbb{P}_{\mathcal{A}}(\pi(\mathbf{s}^t)^{a_1}, a_2 \dots a_n) dt$$

where  $c = c(a_1)$ , and as base case we take  $\mathbb{P}_{\mathcal{A}}(\pi(\mathbf{s}), \varepsilon) = 1$ .



# Statistical Model Checking

[FORMATS11,  
LPAR12, RV12]





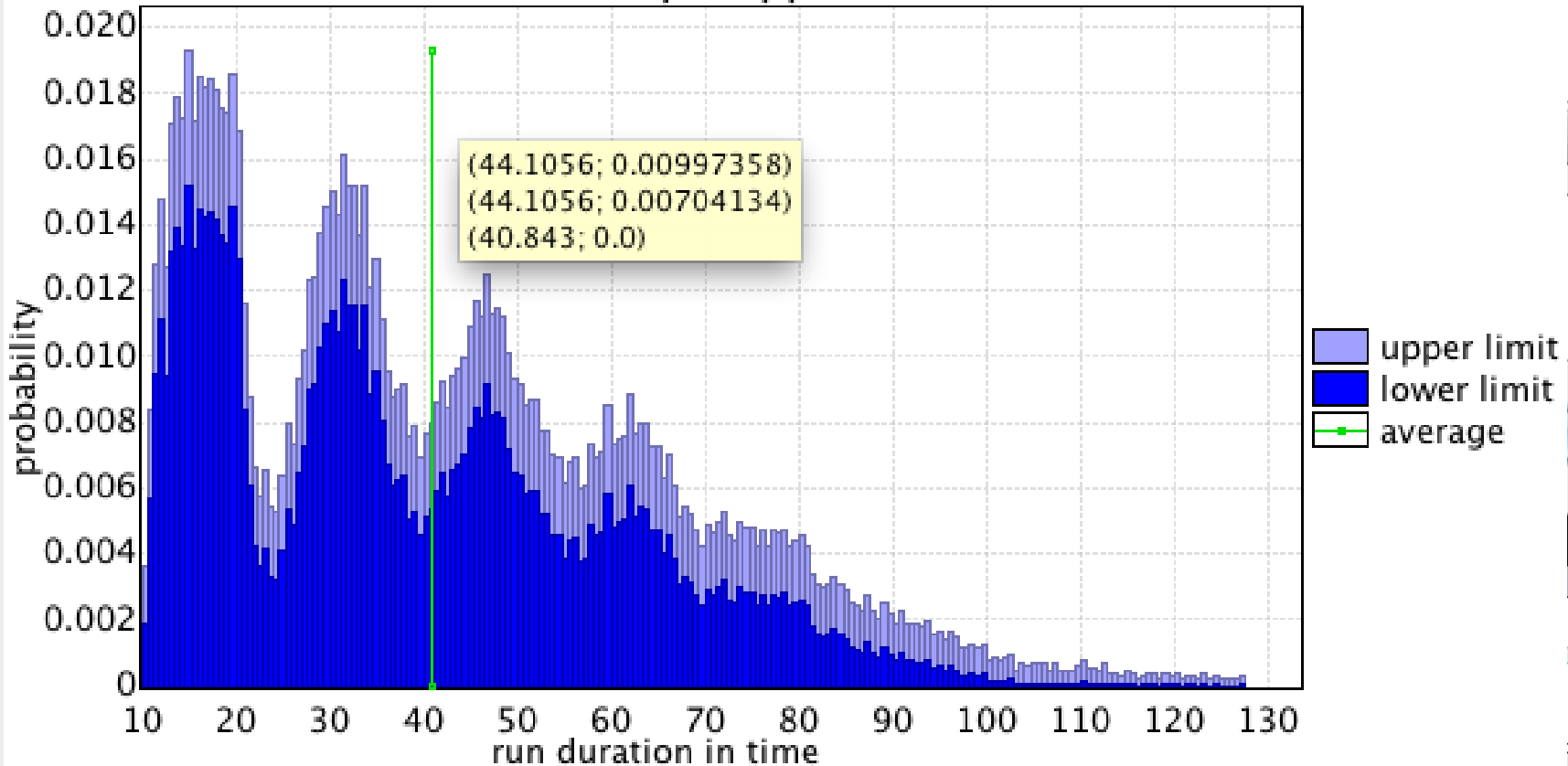
# Queries in UPPAAL SMC

Pr[  $\leq 200$ ]( $\langle \rangle$  Train(5).Cross)

Message

Pr[  $\leq 200$ ]( $\langle \rangle$  Train(5).Cross)

Probability Clopper-Pearson CIs



Parameters:  $\alpha=0.01$ ,  $\epsilon=0.01$ , bucket width=0.587972, bucket count=200.

Runs: 26492 in total, 26492 displayed, 0 remaining.

Probability sums: 1 displayed, 0 remaining.

Average: 40.843.

OK

x=0

Start  
x  $\leq$  15

!?

Cross

# Queries in UPPAAL SMC

$\text{Pr}[ \leq 100 ] ( \langle \rangle \text{Train}(0).\text{Cross} ) \geq 0.8$

The screenshot shows the UPPAAL SMC interface. On the left, the 'Enabled Transitions' list includes 'Train(5)' and 'appr[0]: Train(0) --> Gate'. The central panel displays the model's state with variables like 'Gate.list[0] = 5', 'Gate.len = 5', and 'Train(0).x >= 23'. A 'Message' dialog box is open, displaying the result: '(149 runs) H1: Pr(<> ...) <= 0.79 with confidence 0.99.' An 'OK' button is visible at the bottom of the dialog.

$\text{Pr}[ \leq 100 ] ( \langle \rangle \text{Train}(0).\text{Cross} ) \geq 0.5$

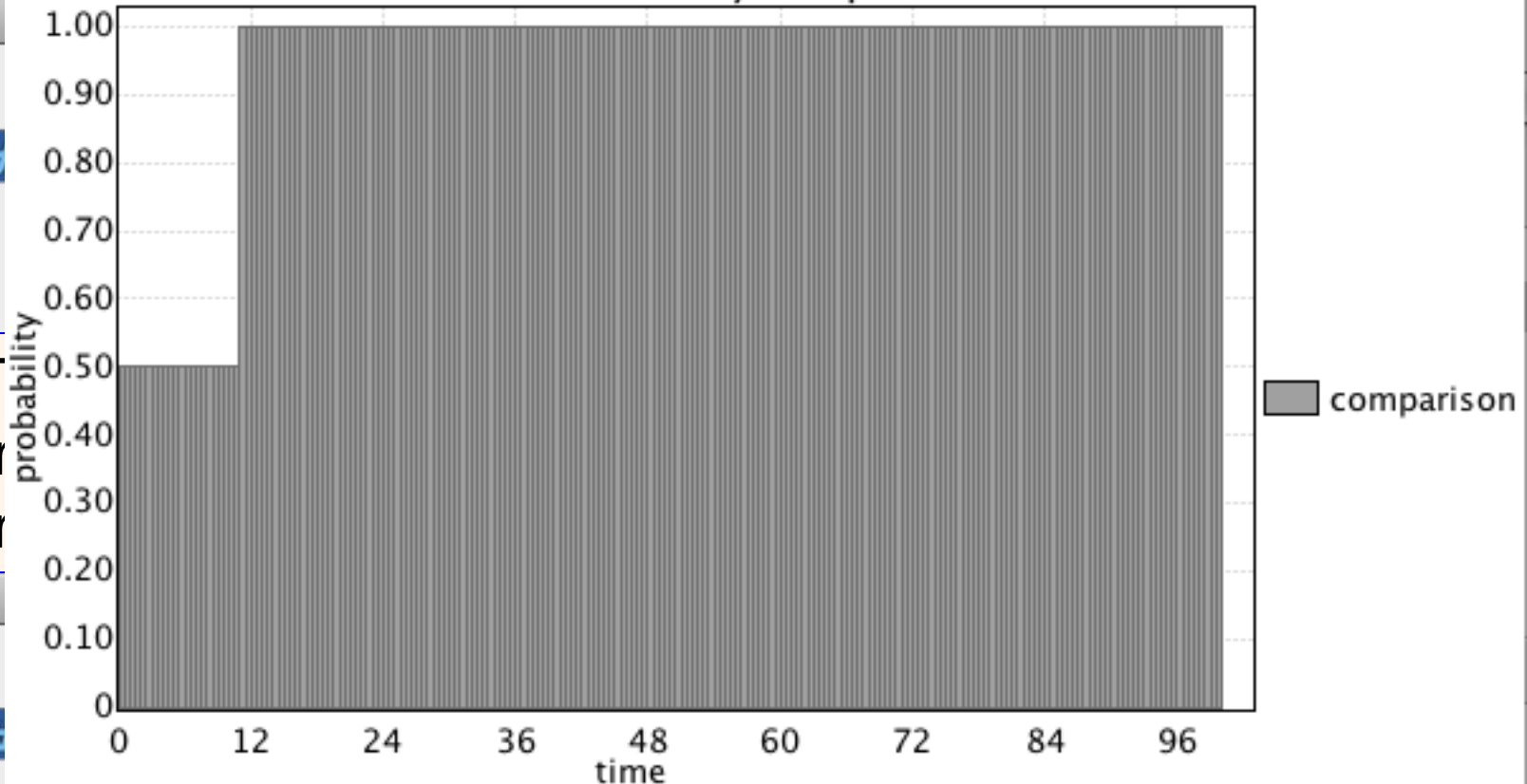
The screenshot shows the UPPAAL SMC interface with a different query. The 'Enabled Transitions' list includes 'appr[3]: Train(5) --> Gate' and 'appr[2]: Train(2) --> Gate'. The central panel shows state variables like 'Train(1).x <= Train(0).x' and 'Train(5).x - Train(0).x <'. A 'Message' dialog box is open, displaying the result: '(651 runs) H0: Pr(<> ...) >= 0.51 with confidence 0.99.' An 'OK' button is visible at the bottom of the dialog. Below the dialog, the state of 'Train(4)' and 'Train(5)' is visible, showing 'x >= 3' and 'leave[4]!'.

# Queries in UPPAAL SMC

$\text{Pr}[\leq 100](\langle \rangle \text{Train}(5).\text{Cross}) \geq$

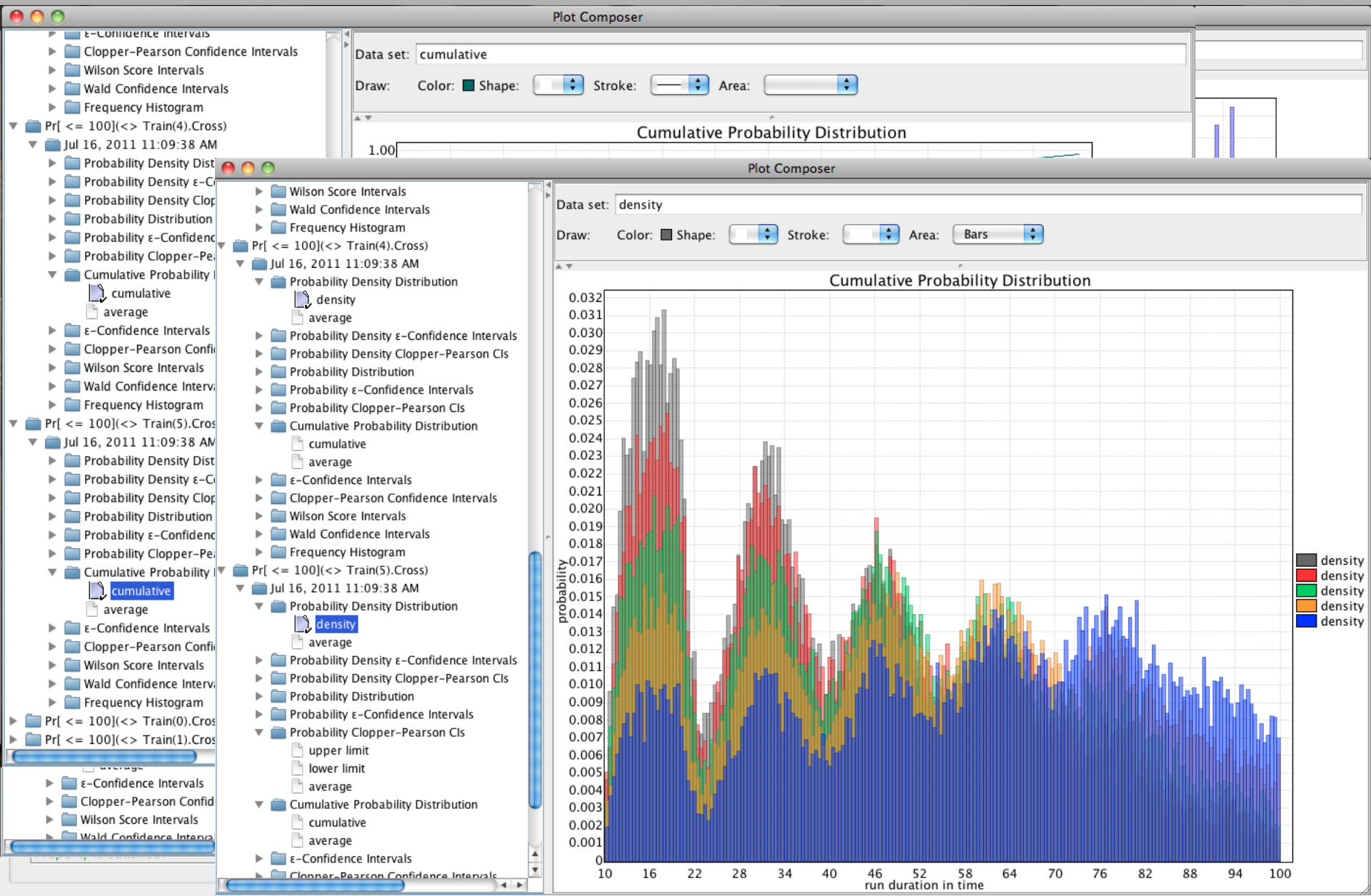
$\text{Pr}[\leq 100](\langle \rangle \text{Train}(1).\text{Cross})$

Probability comparison



value 0.0 means less-than is true.  
value 0.5 means probabilities are indistinguishable.  
value 1.0 means greater-than is true.

# Analysis Tool: Plot Composer

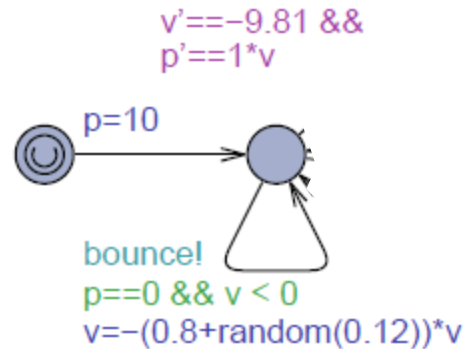


# Demo



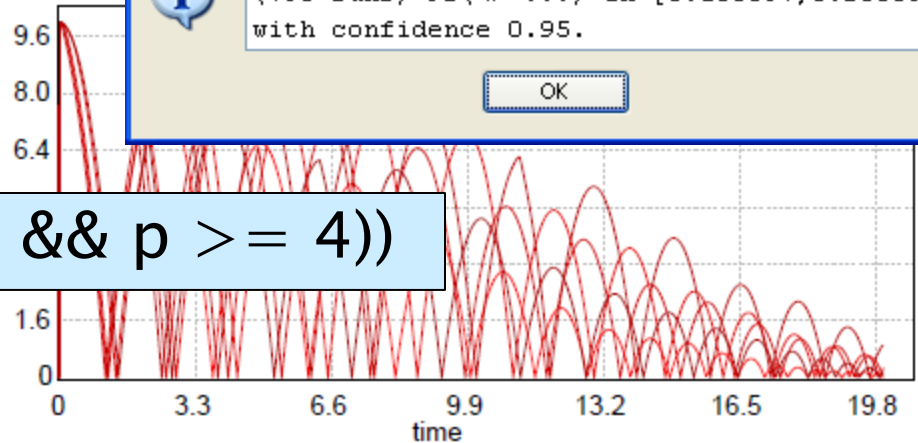
# Stochastic Hybrid Systems

## ■ A Bouncing Ball



Simulate 5 [ $\leq 20$ ] {p}

$\text{Pr}[\leq 20](\langle \rangle (\text{time} \geq 12 \ \&\& \ p \geq 4))$



# Stochastic Hybrid Systems

- A Bouncing Ball

## UPPAAL SMC

Uniform distributions (bounded delay)

Exponential distributions (unbounded delay)

Syntax for discrete probabilistic choice

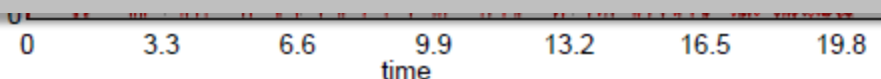
Distribution on next state by use of **random**

Hybrid flow by use of ODEs

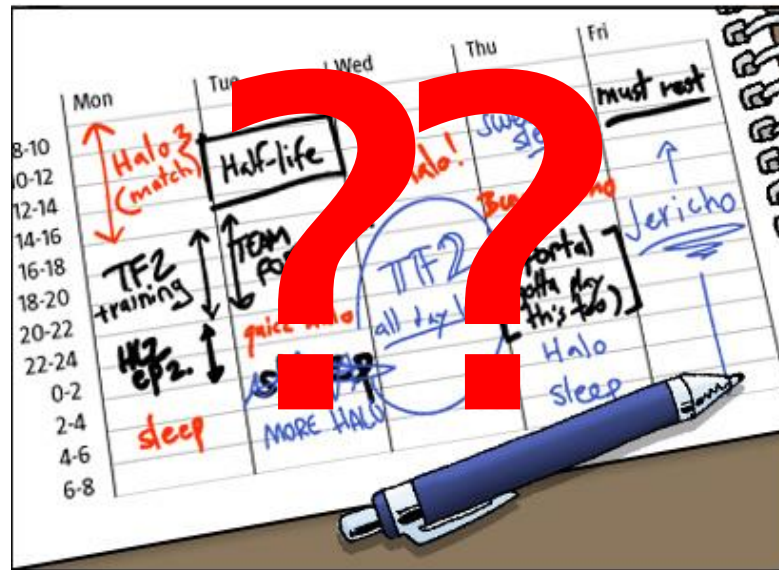
+ usual stuff (structured variables, user-defined types  
user-defined functions, ....)

## Networks

Repeated races between components for outputting



# Schedulability & Performance Analysis

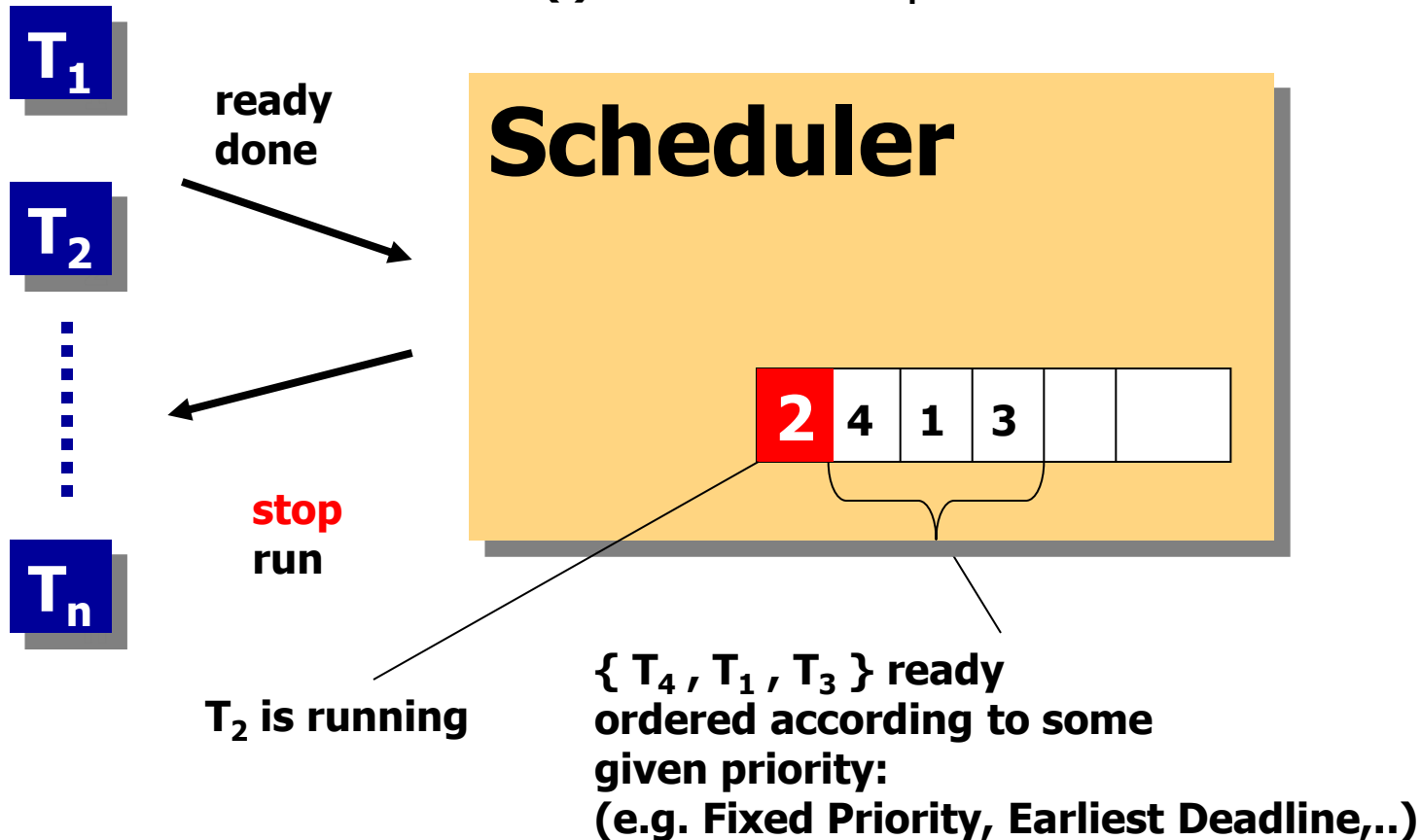




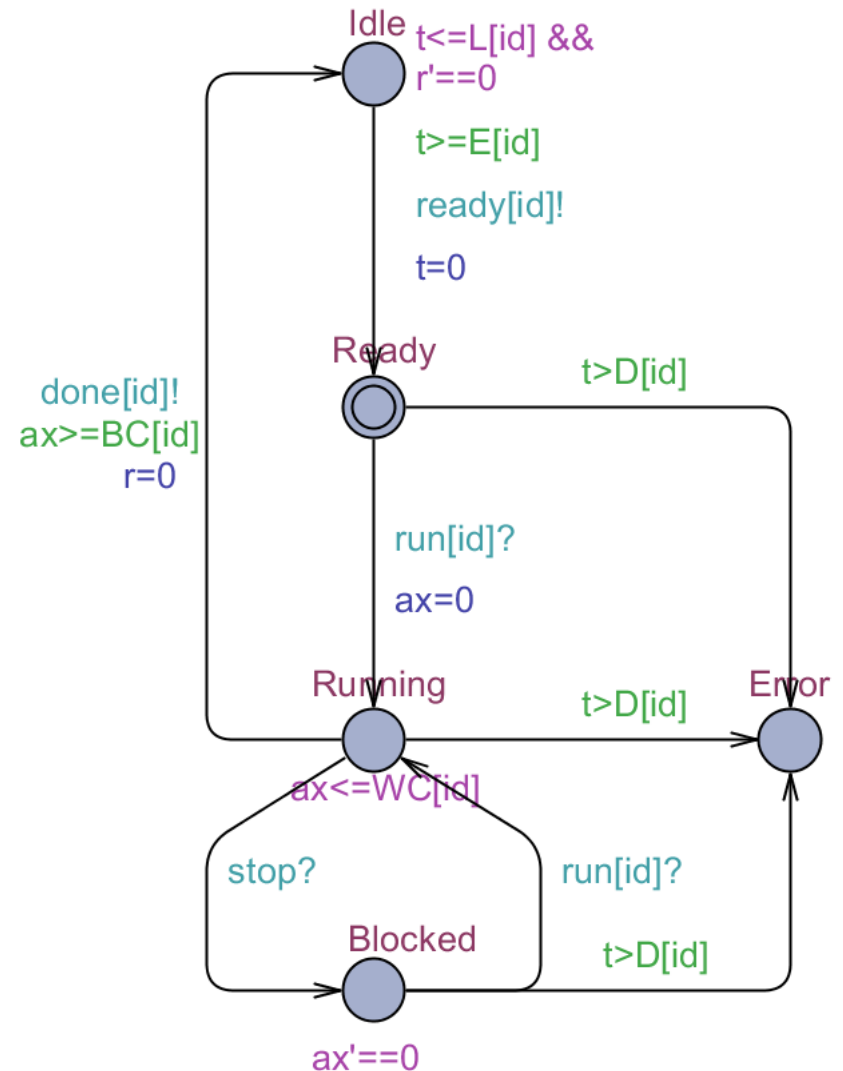
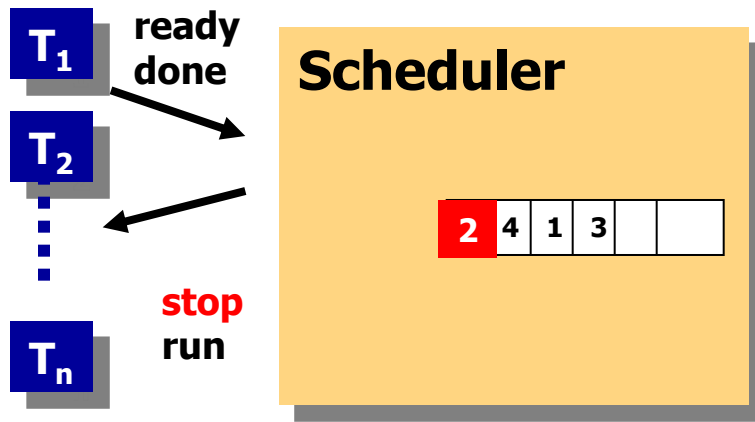
# Task Scheduling

*utilization of CPU*

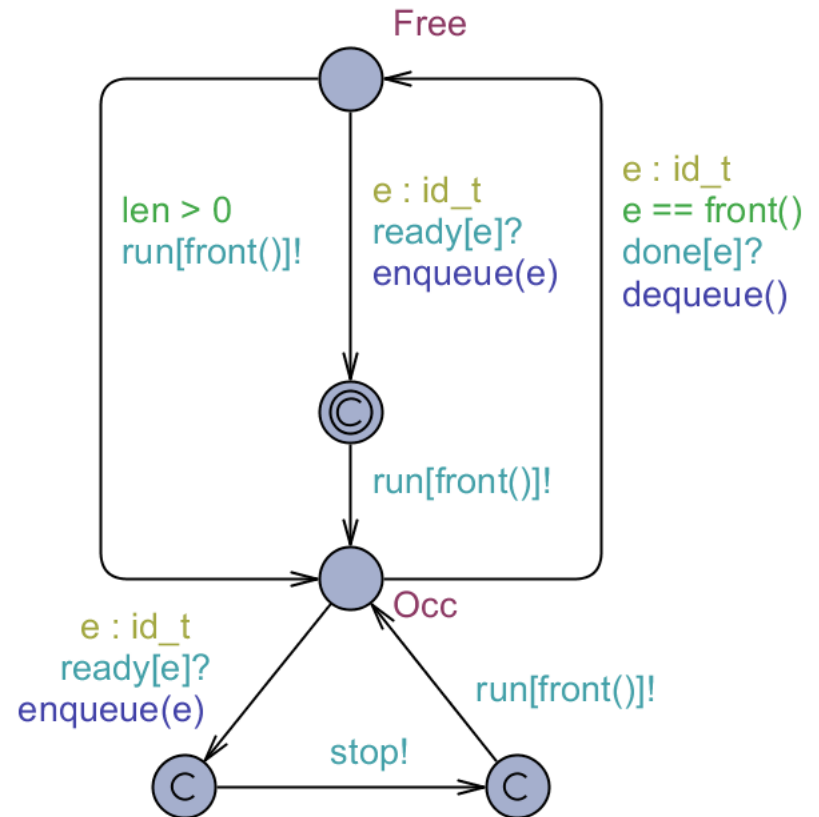
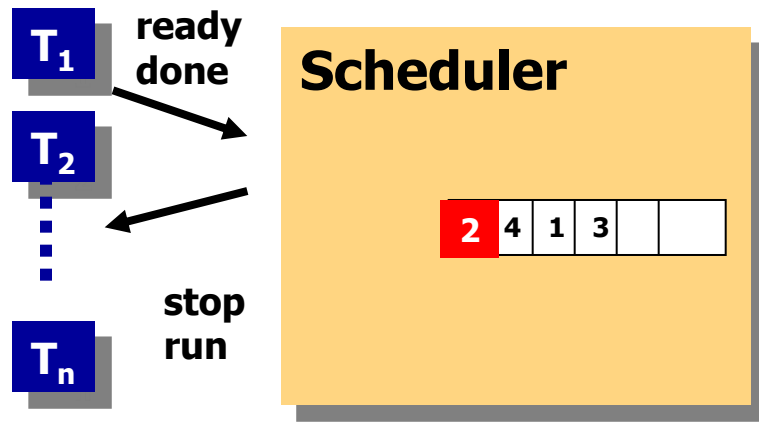
$P(i)$ , **UNI**[ $E(i)$ ,  $L(i)$ ], .. : period or  
earliest/latest arrival or .. for  $T_i$   
 $C(i)$ , **UNI**[ $BC(i)$ ,  $WC(i)$ ] : execution time for  $T_i$   
 $D(i)$ : deadline for  $T_i$



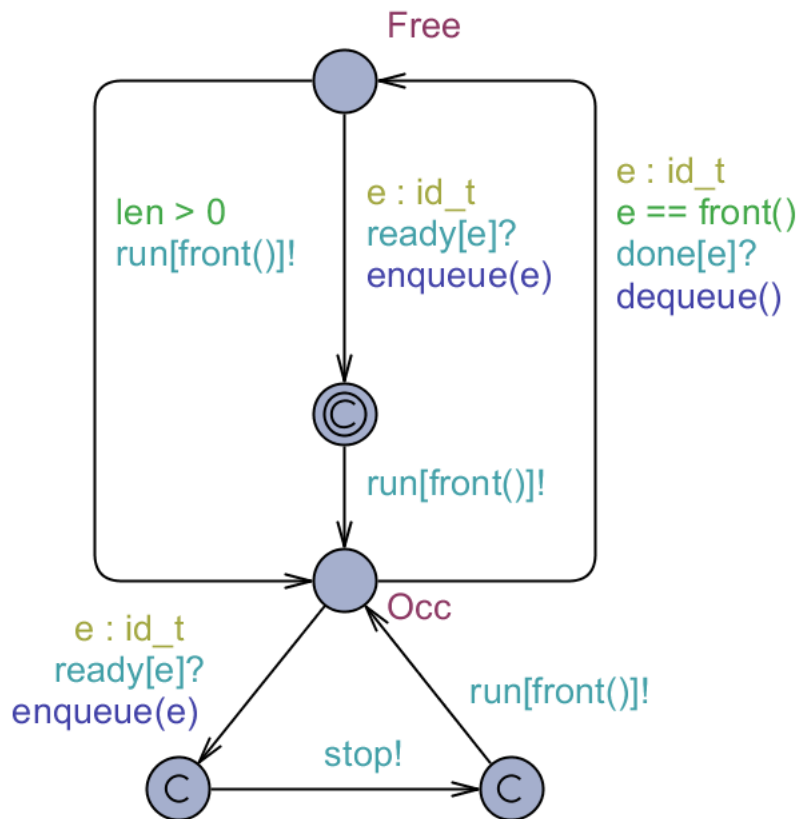
# Modeling Task



# Modeling Scheduler



# Modeling Queue

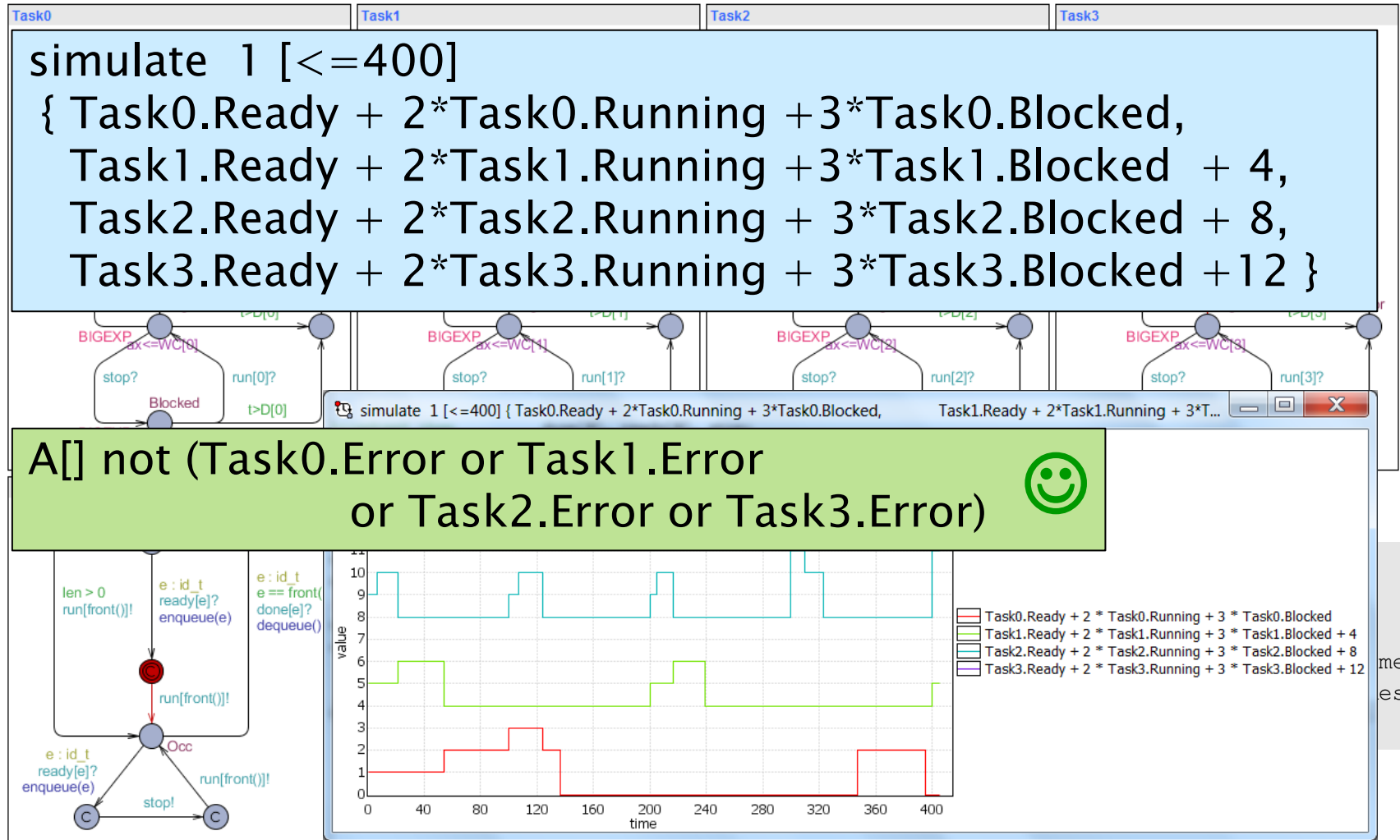


```
// Put an element at the end of the queue
void enqueue(id_t element)
{
  int tmp=0;
  list[len++] = element;
  if (len>0)
  {
    int i=len-1;
    while (i>1 && P[list[i]]>P[list[i-1]])
    {
      tmp = list[i-1];
      list[i-1] = list[i];
      list[i] = tmp;
      i--;
    }
  }
}
```

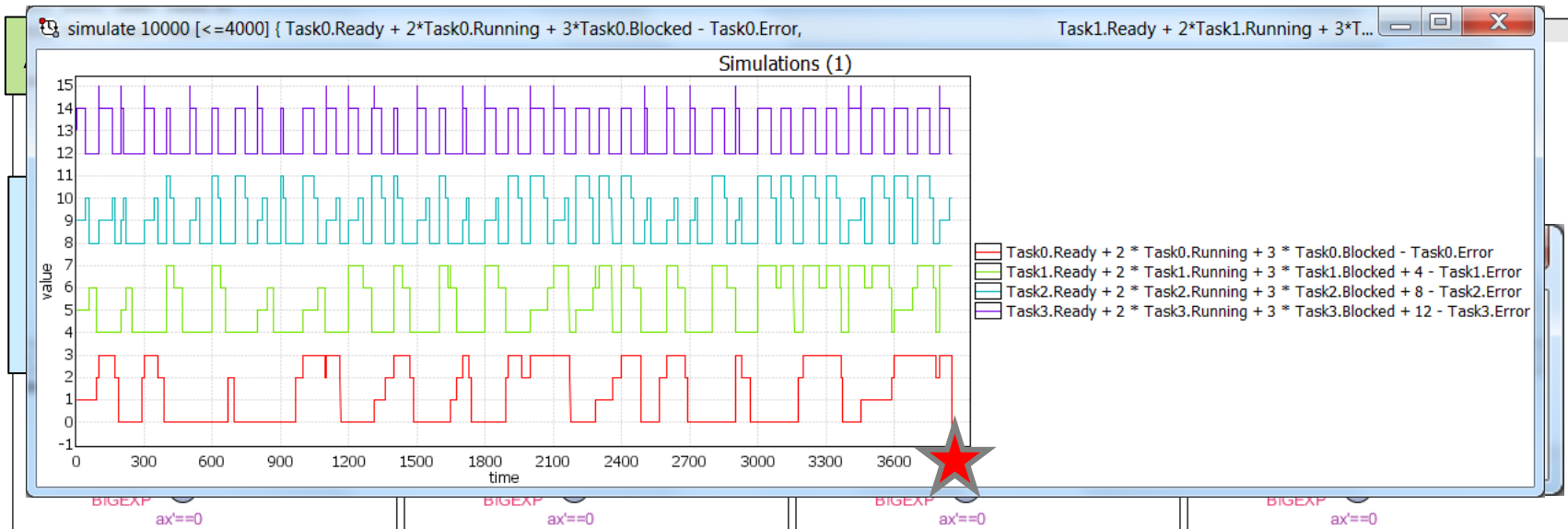
```
// Remove the front element of the queue
void dequeue()
{
  .....
}
```



# Schedulability Analysis

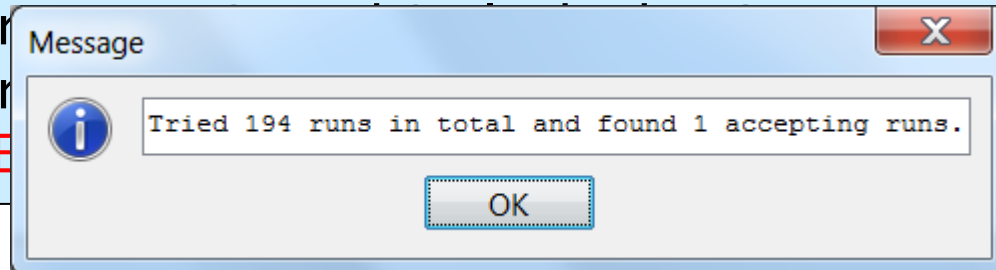


# Schedulability Analysis

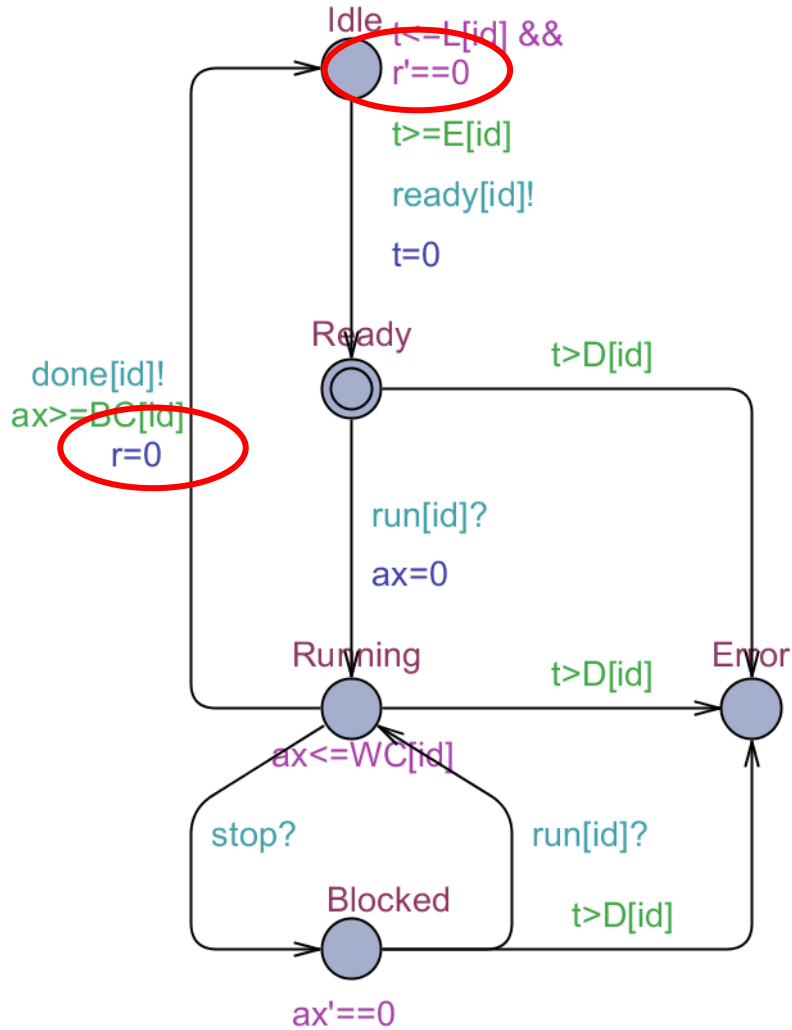


simulate 10000 [<=400]

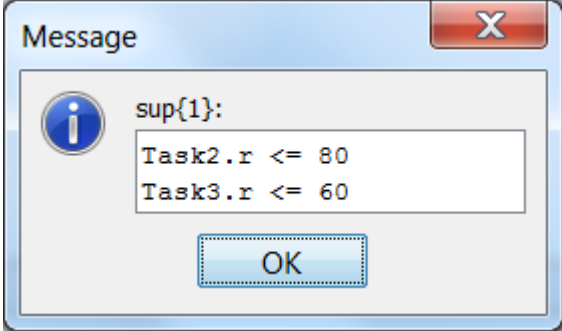
```
{ Task0.Ready + 2*Task0.Running + 3*Task0.Blocked,
  Task1.Ready + 2*Task1.Running + 3*Task1.Blocked + 4,
  Task2.Ready + 2*Task2.Running + 3*Task2.Blocked + 8,
  Task3.Ready + 2*Task3.Running + 3*Task3.Blocked + 12 - Task3.Error
: 1 : (Task0.Error or Task1.Error)
```



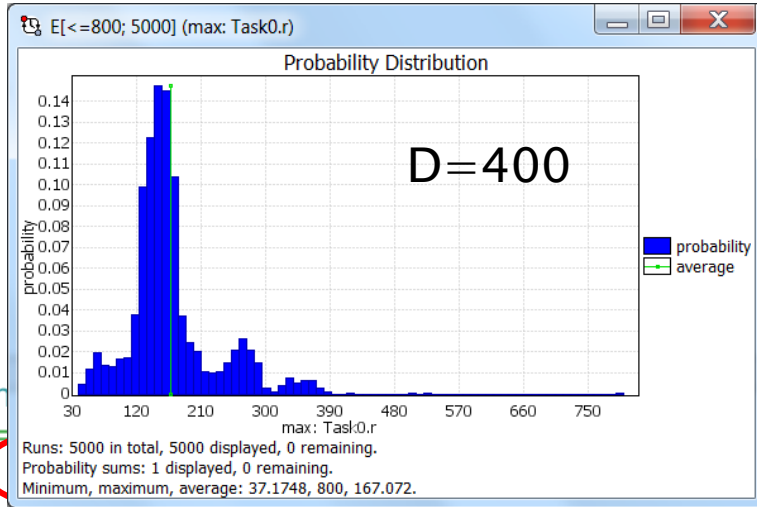
# Performance Analysis



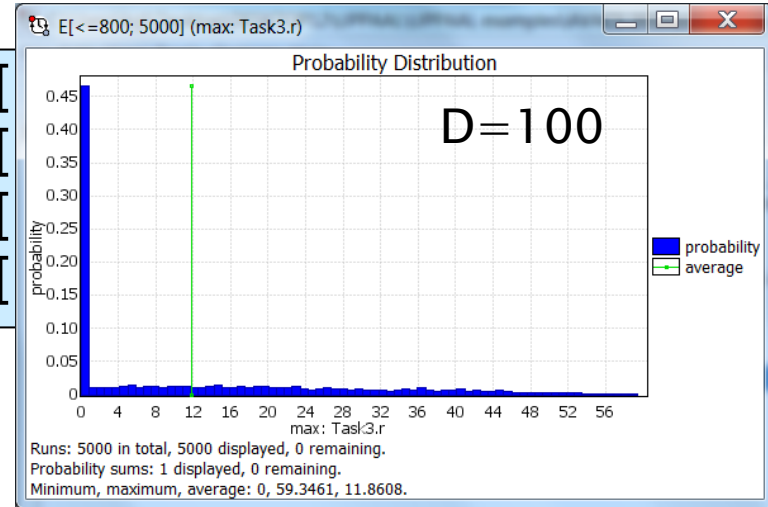
sup : Task2.r, Task3.r



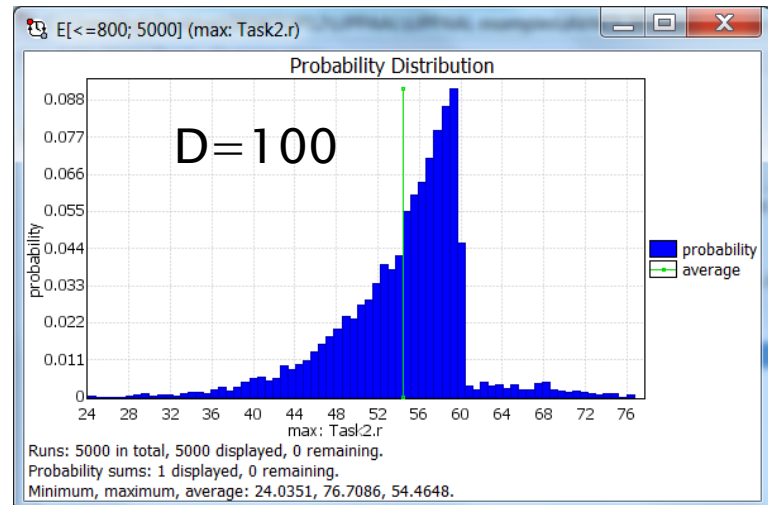
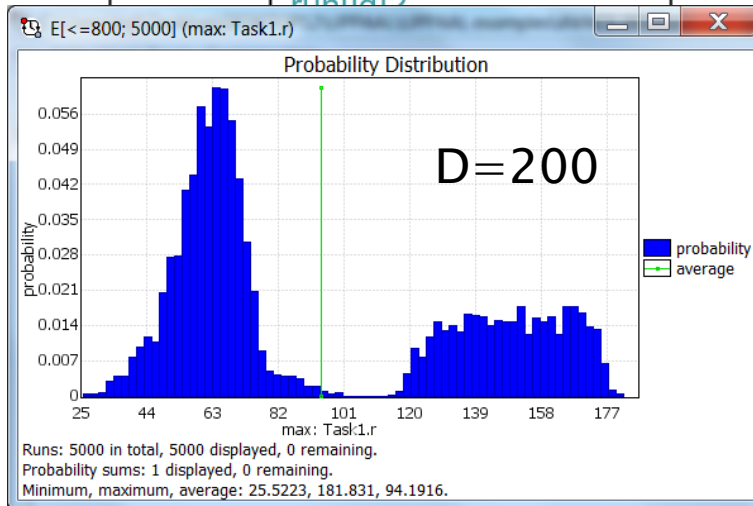
# Performance Analysis



E[  
E[  
E[  
E[



don  
ax>





# Herschel–Planck Scientific Mission at ESA



Attitude and Orbit Control Software

TERMA A/S Steen Ulrik Palm, Jan Storbak Pedersen, Poul Hougaard



# Modeling in UPPAAL

## UPPAAL 4.1 Framework ISoLA 2010

The screenshot displays the UPPAAL 4.1 Framework interface. On the left, there is a 'Transition chooser' with a list of transitions (0.0 to 7.0) and a 'Delay' field set to 13.5. Below it are 'Trace controls' (First, Prev, Play, Next, Last) and a 'Speeder' slider. The 'Simulation Trace' shows a sequence of events: 'initialize: Scheduler --> Bkgnd\_P, NominalE...', 'enqueue: RTEMS\_RTC --> Scheduler', 'Schedule, Idle, Idle, Idle, Idle, Idle, Idle, I...', 'preempt[ctask]: Scheduler --> IdleTask', and '(Preempt, Idle, Idle, Idle, Idle, Idle, I...'. The main area contains four state transition diagrams: 'Scheduler', 'Bkgnd\_P', 'secondF\_2', and 'secondF\_1'. Each diagram shows states (e.g., Idle, Running, Blocked, Reschedule) and transitions with associated guard conditions and actions. The 'Scheduler' diagram shows transitions like 'schedule[task!]', 'release[CPU\_R]?', 'enqueue?', 'cprto[task!]', 'Schedule', and 'Preempt'. The 'Bkgnd\_P' diagram shows transitions like 'starting', 'Idle', 'Running', and 'Error'. The 'secondF\_2' diagram shows transitions like 'Blocked', 'Reschedule', 'WaitForCPU', 'WaitForOther', and 'Handle Pending TCWithBoth'. The 'secondF\_1' diagram shows transitions like 'Blocked', 'Reschedule', 'WaitForCPU', 'DetermineUnit HealthWithSgm\_R', and 'DetermineState'.



# Symbolic MC vs. Statistical MC

Symbolic analysis:

- Preemptive scheduler requires *stop-watches*.
- Exact reachability of stop-watch automata is *undecidable*.
- UPPAAL provides *over-approximation* for stop-watches.
- $\Rightarrow$  symbolic analysis may give spurious errors, but still suitable for *proving safety/schedulability*.

Statistical analysis:

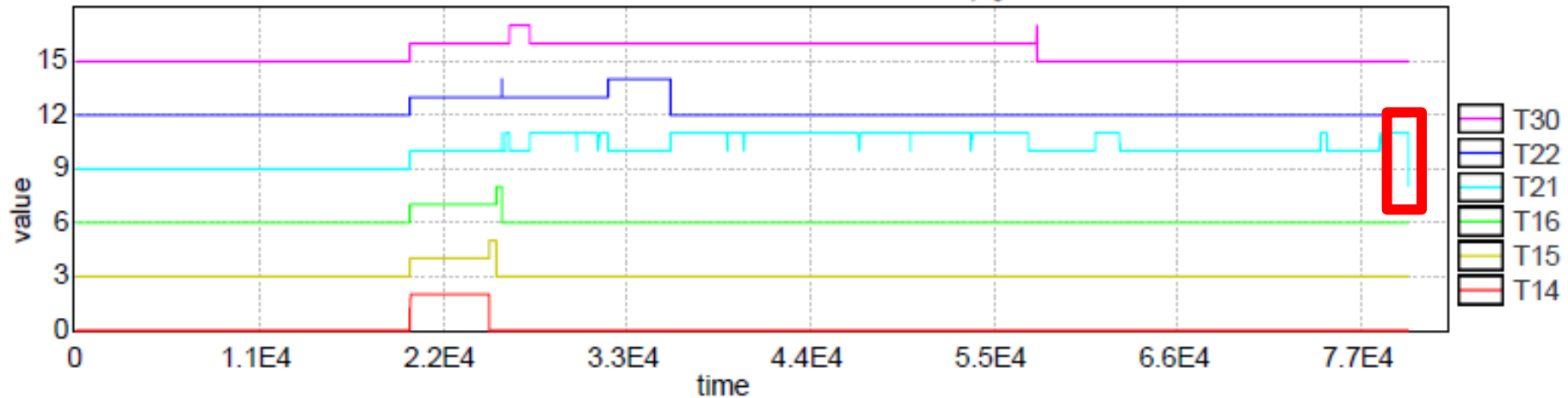
- can show *presence of errors* but not absence.
- $\Rightarrow$  suitable for *disproving schedulability*.

$f = \text{BCET}/\text{WCET}$ :	0-71%	72-86%	87-89%	90-100%
Symbolic MC:	maybe	maybe	n/a	<b>Safe</b>
Statistical MC:	<b>Unsafe</b>	maybe	maybe	maybe



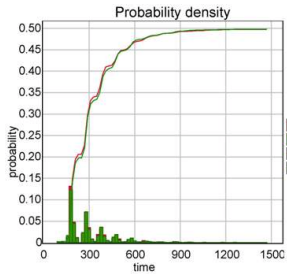
# SMC Simulation to Find Error

Herschel deadline violation with  $f = 50\%$ :

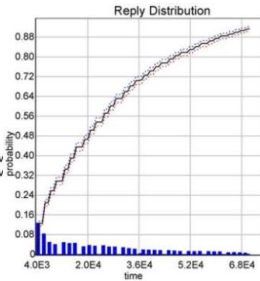


```
simulate 10000 [ <= 300 ] {  
  (T(1).Ready + T(1).Computing + T(1).Release + runs[1] - 2 * T(1)  
  (T(2).Ready + T(2).Computing + T(2).Release + runs[2] - 2 * T(1)  
  (T(3).Ready + T(3).Computing + T(3).Release + runs[3] - 2 * T(1)  
} : 1 : error
```

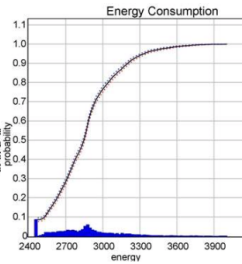
# Other Case Studies



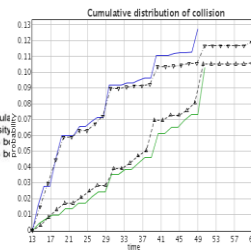
FIREWIRE



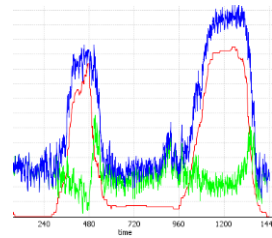
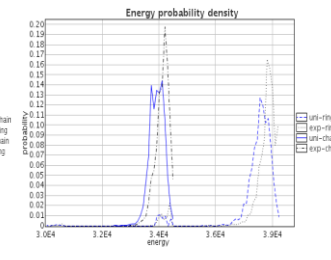
BLUETOOTH



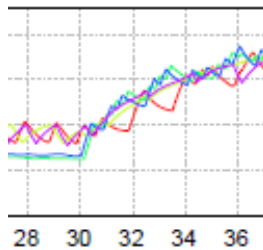
10 node LMAC



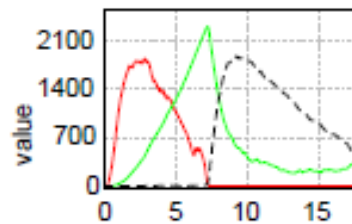
Schedulability Analysis for Mix Cr Sys



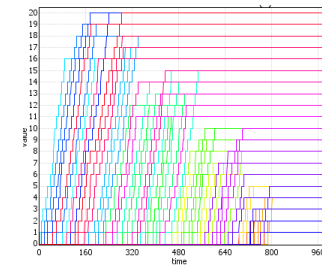
Smart Grid Demand / Response



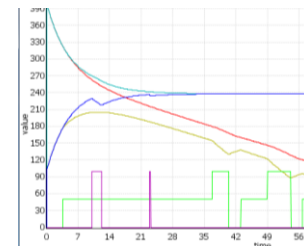
Energy Aware Buildings



Genetic Oscillator (HBS)



Passenger Seating in Aircraft



**Battery Scheduling (SENSATION)**  
Erik Wogensen



# SMC Queries – Examples

- $\text{Pr}[\leq 100](\langle \rangle \text{ goal})$
- $\text{Pr}[\#\leq 10]([] \text{ safe})$
- $\text{Pr}[x\leq 200](\langle \rangle \text{ goal}) \geq 0.3$
- $E[\leq 100; 1000](\text{min: expr})$
- `simulate 10 [ $\leq 100$ ] { e1, e2, x1 }`
- `simulate 100 [ $\leq 10$ ] { e } : 2 : goal`

*Exercise 28 (Jobshop scheduling part 2)*

