

Statistical Model Checking in UPPAAL

Alexandre David, Kim G. Larsen,
Marius Mikucionis

Axel Legay, 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 ✓

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

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

Reachability ✓

Liveness ✓

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

$A \leftrightarrow \dots E[] \dots$ ✓

Limited quantitative analysis ✓

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 ✓

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

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

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

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

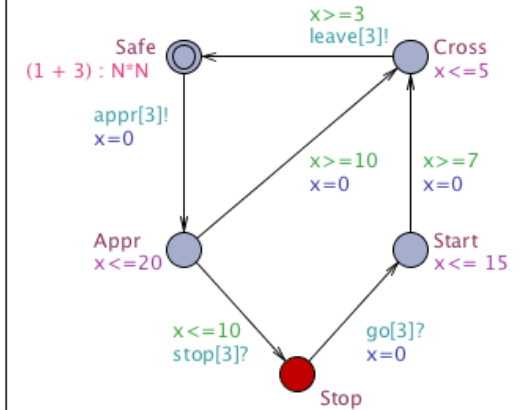
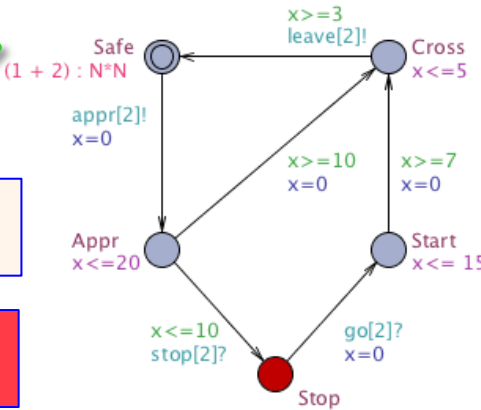
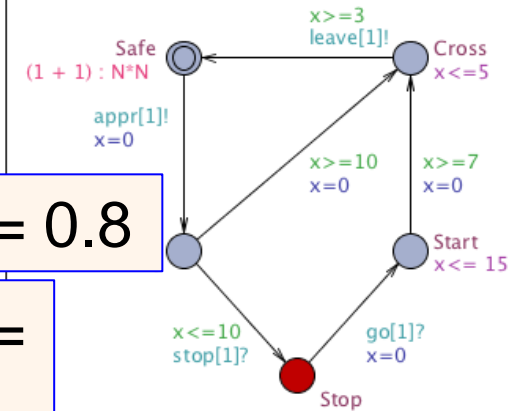
State-space explosion ✓

Generate runs

Performance properties

State-space explosion

Editor Simulator Verifier



Overview

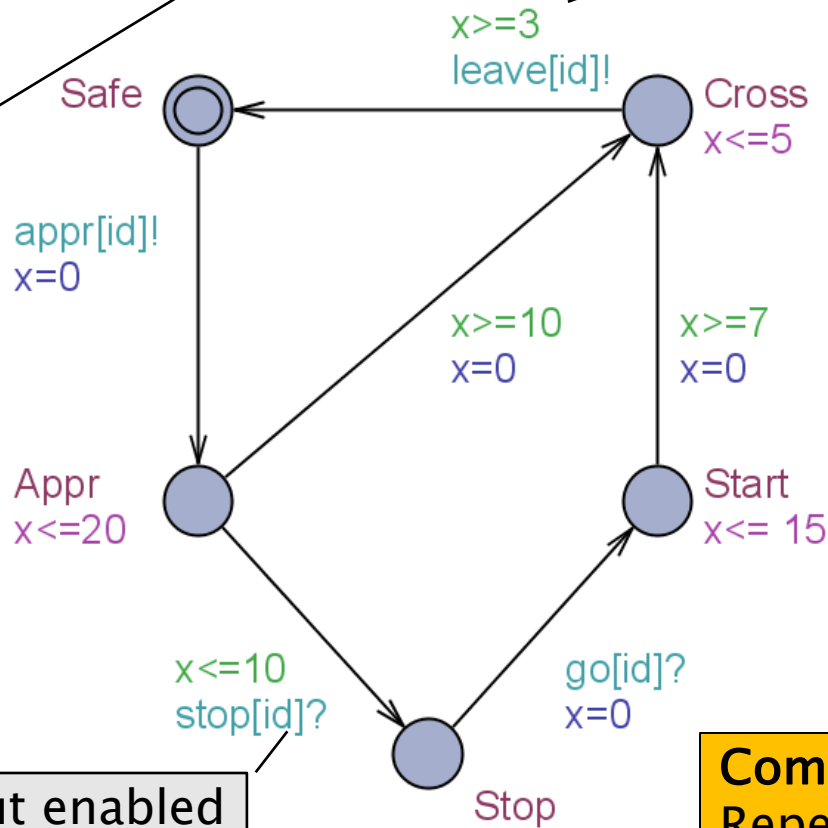
- Stochastic Semantics of Networks of Timed Automata
- Statistical Model Checking in UPPAAL
 - Estimation
 - Sequential Hypothesis Testing
 - Sequential Probability Comparison
 - Parameterized Probability Comparison
- SMC of Hybrid Automata
- Case Studies & Demo



Stochastic Semantics of TA

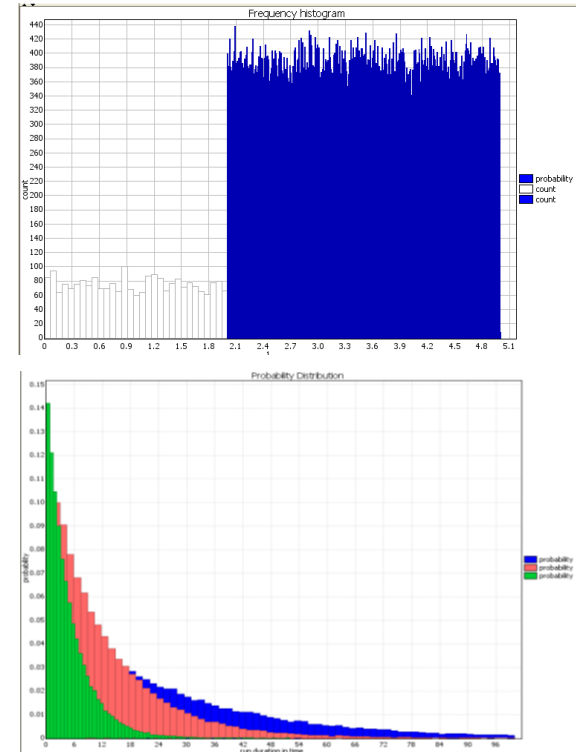
Exponential Distribution

Uniform Distribution

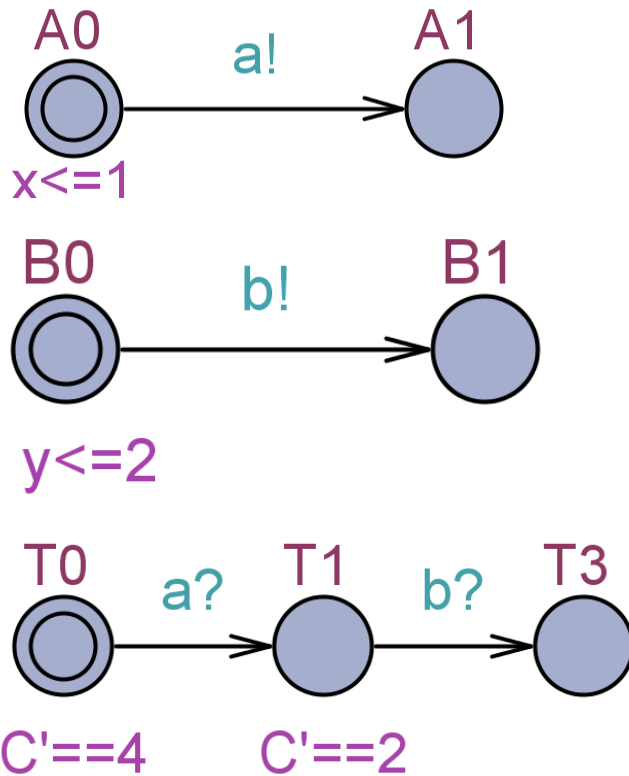


Input enabled

Composition =
Repeated races between components



Stochastic Semantics of Timed Automata

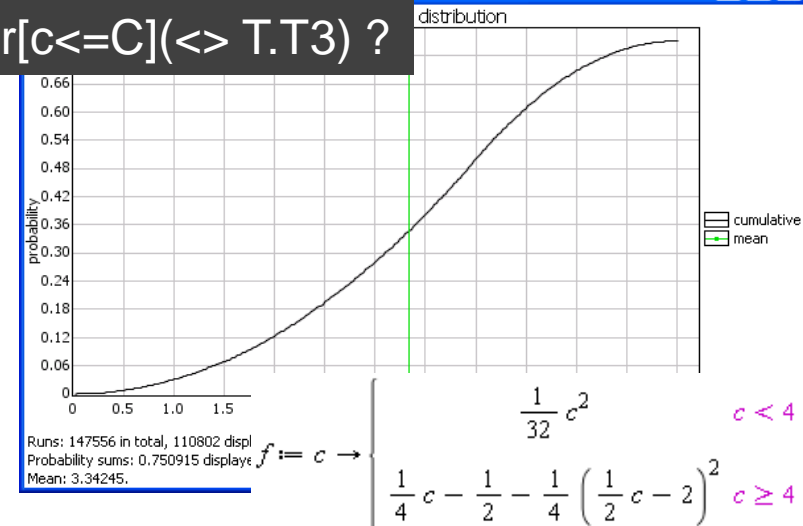


Composition = Race between components for outputting

$\Pr[\text{time} \leq T](\langle \rangle T.T3) ?$

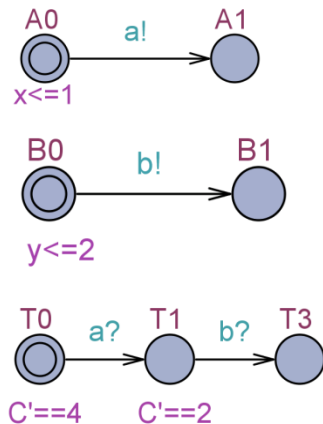


$\Pr[c \leq C](\langle \rangle T.T3) ?$



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

$t_1 a_1 t_2 a_2 \dots t_n a_n$

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

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

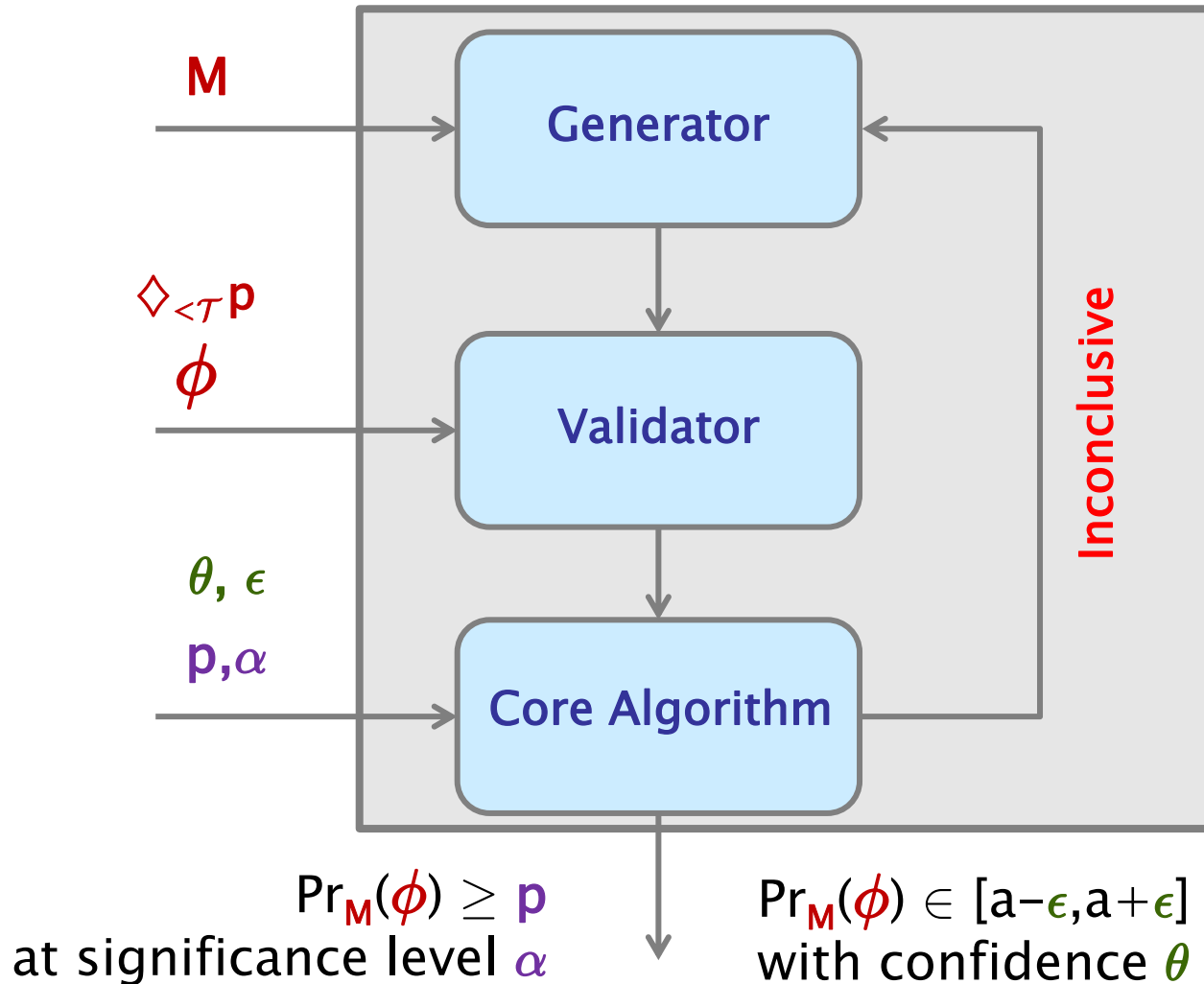
$$\int_{t \geq 0} \mu_{s_c}(t) \cdot \left(\prod_{j \neq c} \int_{\tau > t} \mu_{s_j}(\tau) d\tau \right) \cdot \gamma_{s_c t}(a_1) \cdot \mathbb{P}_{\mathcal{A}}(\pi(\mathbf{s}^t)^{a_1}, a_2 \dots a_n) dt$$

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]



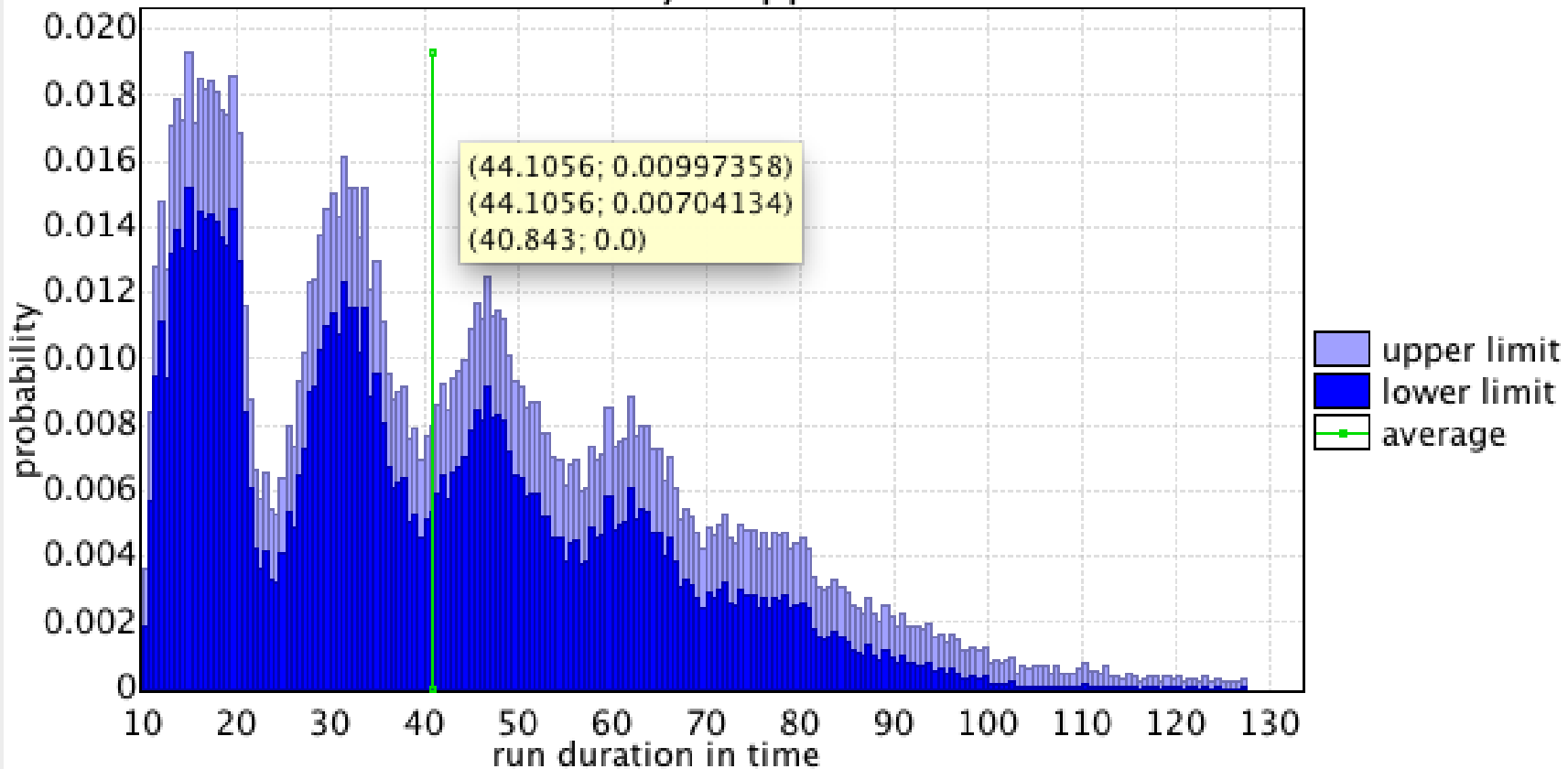
Queries in UPPAAL SMC

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

Message

Pr[≤ 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 runs: 1 displayed, 0 remaining.

Average: 40.843.

Kim Larsen [9]

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, there are panels for 'Enabled Transitions' and 'Trace File'. The 'Enabled Transitions' panel shows 'Train(5)' and 'appr[0]: Train(0) --> Gate'. The 'Trace File' panel shows a sequence of transitions: (Safe, Stop, Safe, Appr, Stop, Start, Stopping), stop[tail0]: Gate --> Train(3), (Safe, Stop, Safe, Stop, Stop, Start, Occ), appr[2]: Train(2) --> Gate, (Safe, Stop, Appr, Stop, Stop, Start, Stopping), stop[tail0]: Gate --> Train(2), and (Safe, Stop, Stop, Stop, Stop, Start, Occ). The 'Trace File' panel also has buttons for 'Prev', 'Next', 'Replay', 'Open', 'Save', and 'Random', and a speed slider from 'Slow' to 'Fast'. The main window displays a list of variables: Gate.list[0] = 5, Gate.list[1] = 1, Gate.list[2] = 4, Gate.list[3] = 3, Gate.list[4] = 2, Gate.list[5] = 0, Gate.list[6] = 0, Gate.len = 5, Train(0).x >= 23, Train(1).x ∈ [13,60], and Train(2).x ∈ [0,15]. A 'Message' dialog box is open, displaying the result: '(149 runs) H1: Pr(<> ...) <= 0.79 with confidence 0.99.' and an 'OK' button.

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

The screenshot shows the UPPAAL SMC interface. On the left, there are panels for 'Enabled Transitions' and 'Trace File'. The 'Enabled Transitions' panel shows 'appr[3]: Train(5) --> Gate', (Safe, Stop, Safe, Appr, Stop, Start, Stopping), stop[tail0]: Gate --> Train(3), (Safe, Stop, Safe, Stop, Stop, Start, Occ), appr[2]: Train(2) --> Gate, (Safe, Stop, Appr, Stop, Stop, Start, Stopping), stop[tail0]: Gate --> Train(2), and (Safe, Stop, Stop, Stop, Stop, Start, Occ). The 'Trace File' panel also has buttons for 'Prev', 'Next', 'Replay', 'Open', 'Save', and 'Random', and a speed slider from 'Slow' to 'Fast'. The main window displays a list of variables: Train(1).x <= Train(0).x, Train(1).x - Train(5).x ∈, Train(2).x - Train(3).x ∈, Train(3).x - Train(5).x ∈, Train(4).x - Train(0).x <, Train(1).x - Train(4).x ∈, Train(4).x - Train(5).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.' and an 'OK' button. Below the dialog box, there are two 'Train' panels: 'Train(4)' and 'Train(5)'. 'Train(4)' shows 'x >= 3 leave[4]!' and 'Cross'. 'Train(5)' shows 'x >= 3 leave[5]!' and 'Cross'. There are also 'Stop' buttons for each train.

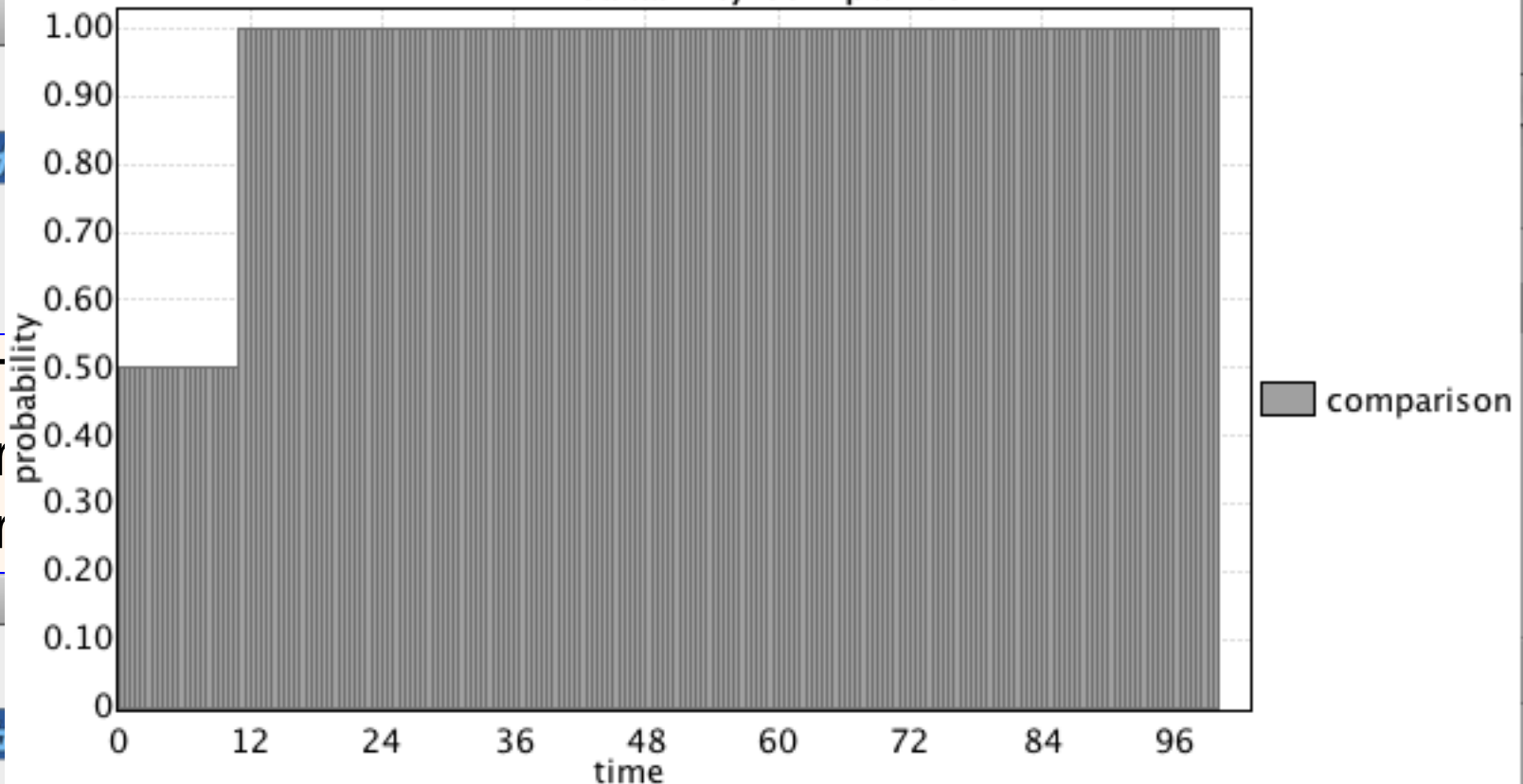


Queries in UPPAAL SMC

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

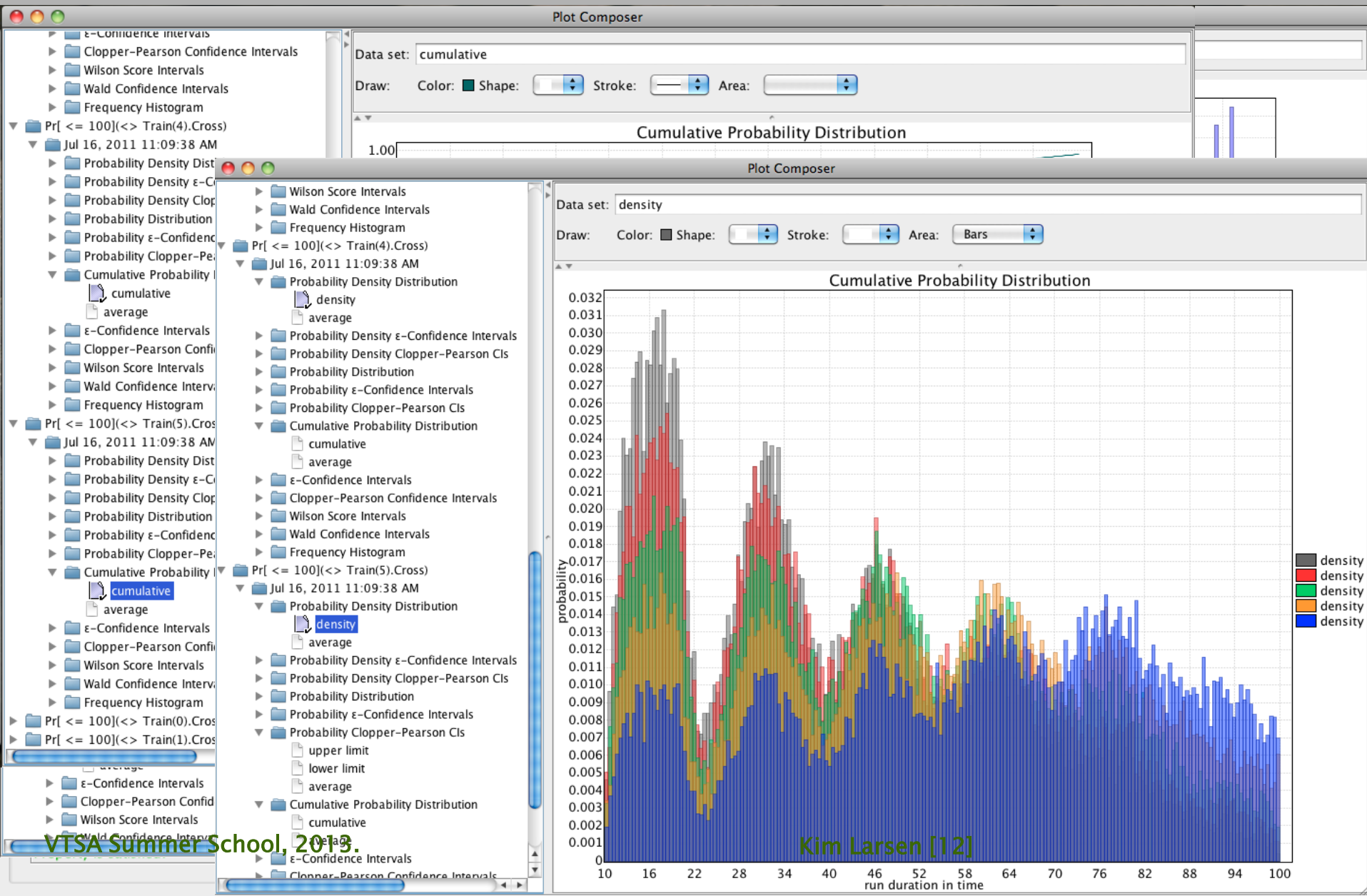
$\text{Pr}[\leq 100](\langle \rangle \text{Train}(5).\text{Cross}) \geq \text{Pr}[\leq 100](\langle \rangle \text{Train}(1).\text{Cross})$

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.

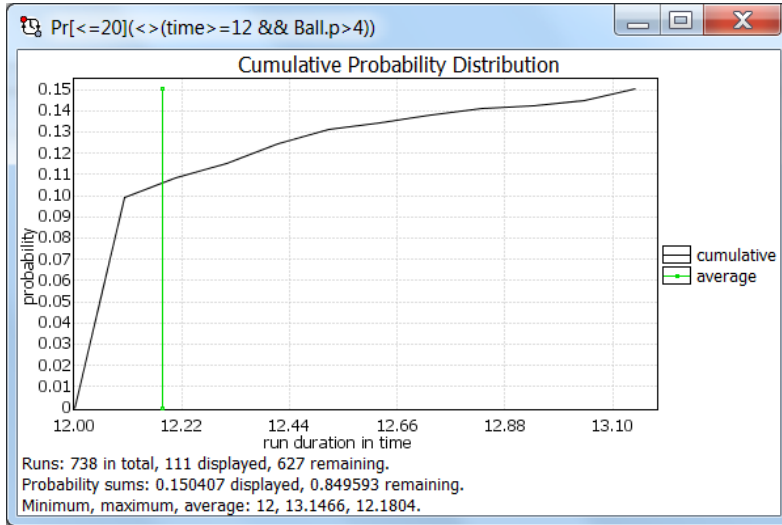
Analysis Tool: Plot Composer



Demo



Stochastic Hybrid Systems



Ball

&

0.12)) * v

Player 1

Message

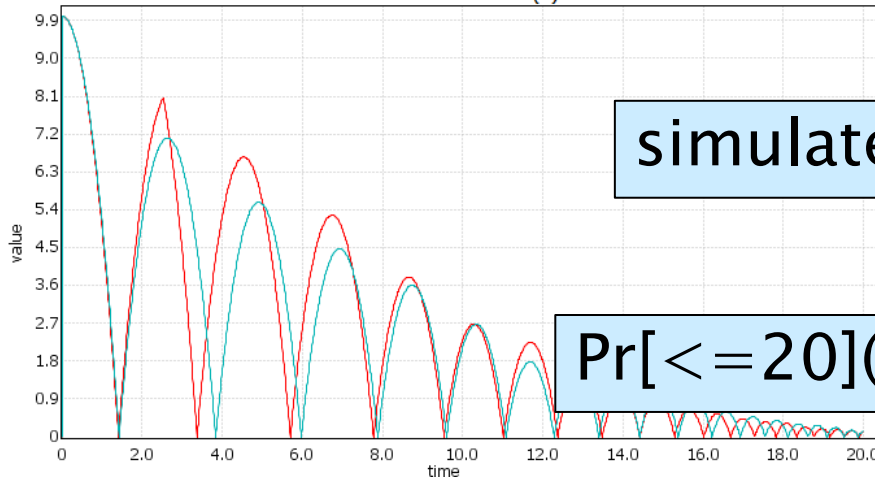
(738 runs) Pr(<> ...) in [0.100407, 0.200407] with confidence 0.95.

OK

x=0



x<=3



simulate 1 [≤ 20]{Ball1.p, Ball2.p}

Pr[≤ 20]($\langle \rangle$ (time ≥ 12 && Ball.p > 4))



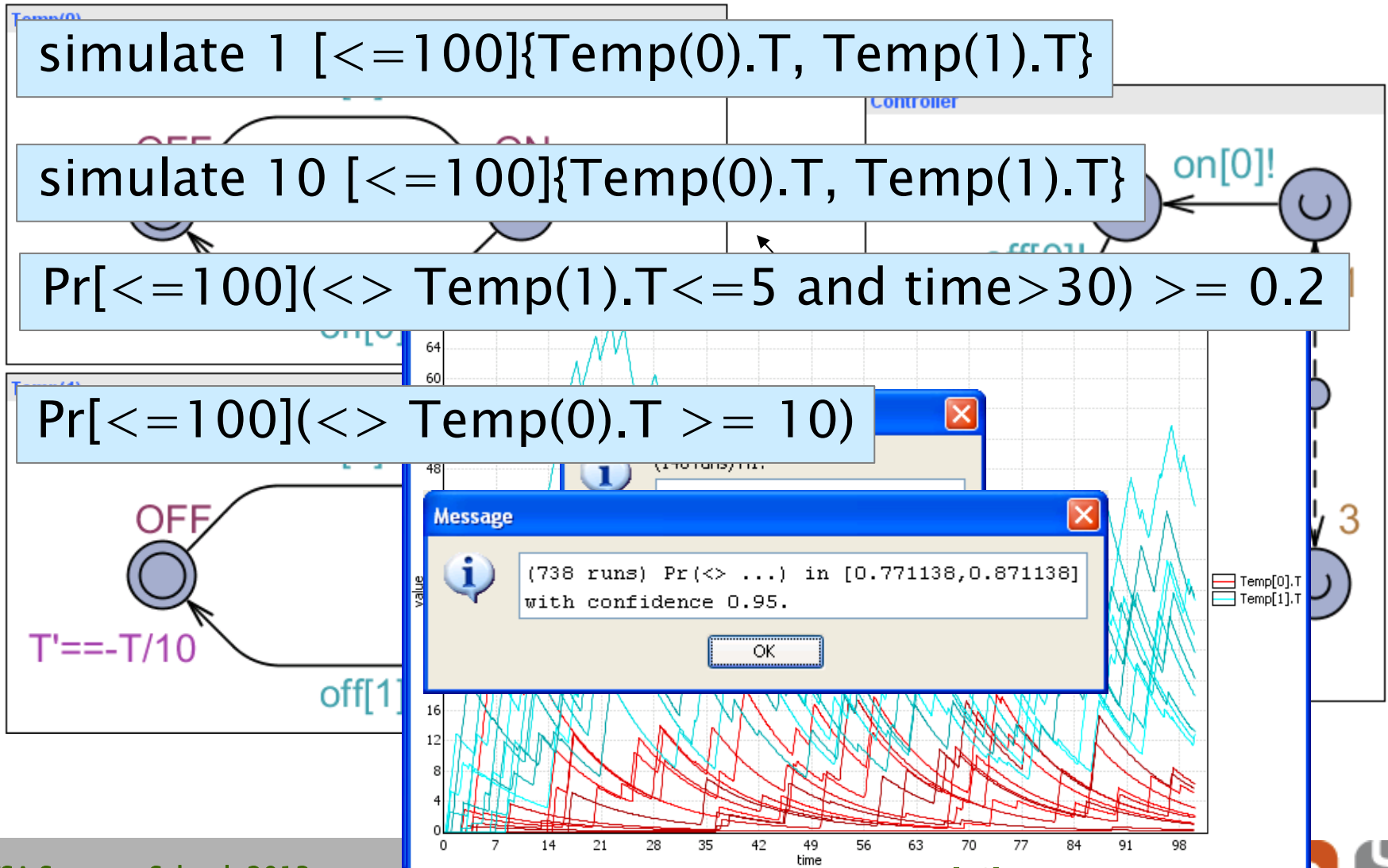
Stochastic Hybrid Systems

simulate 1 [≤ 100]{Temp(0).T, Temp(1).T}

simulate 10 [≤ 100]{Temp(0).T, Temp(1).T}

$\Pr[\leq 100](\langle \rangle \text{Temp(1).T} \leq 5 \text{ and time} > 30) \geq 0.2$

$\Pr[\leq 100](\langle \rangle \text{Temp(0).T} \geq 10)$



Stochastic Hybrid Systems

- A Bouncing Ball

UPPAAL SMC

Uniform distributions (bounded delay)

Exponential distributions (unbounded delay)

Syntax for discrete probabilistic choice

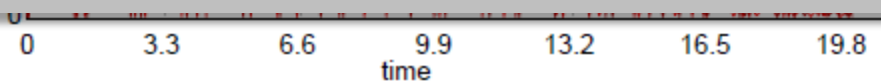
Distribution on next state by use of **random**

Hybrid flow by use of ODEs

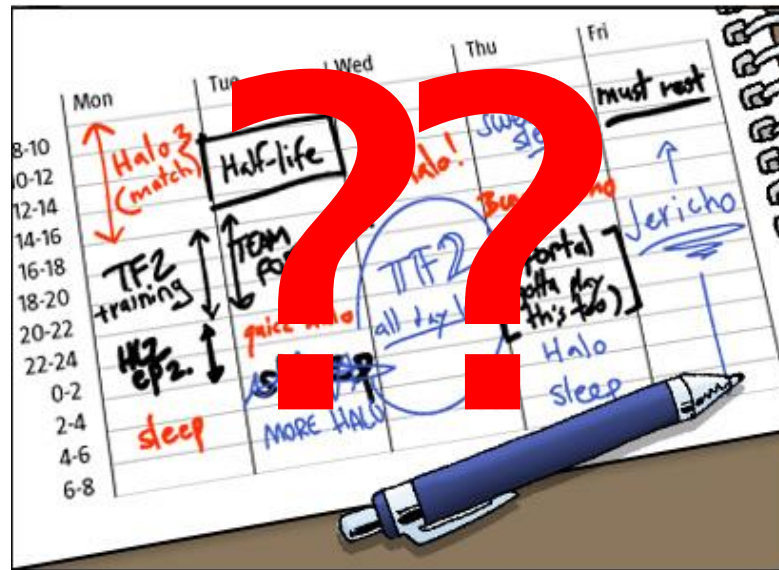
+ usual stuff (structured variables, user-defined types
user-defined functions,)

Networks

Repeated races between components for outputting



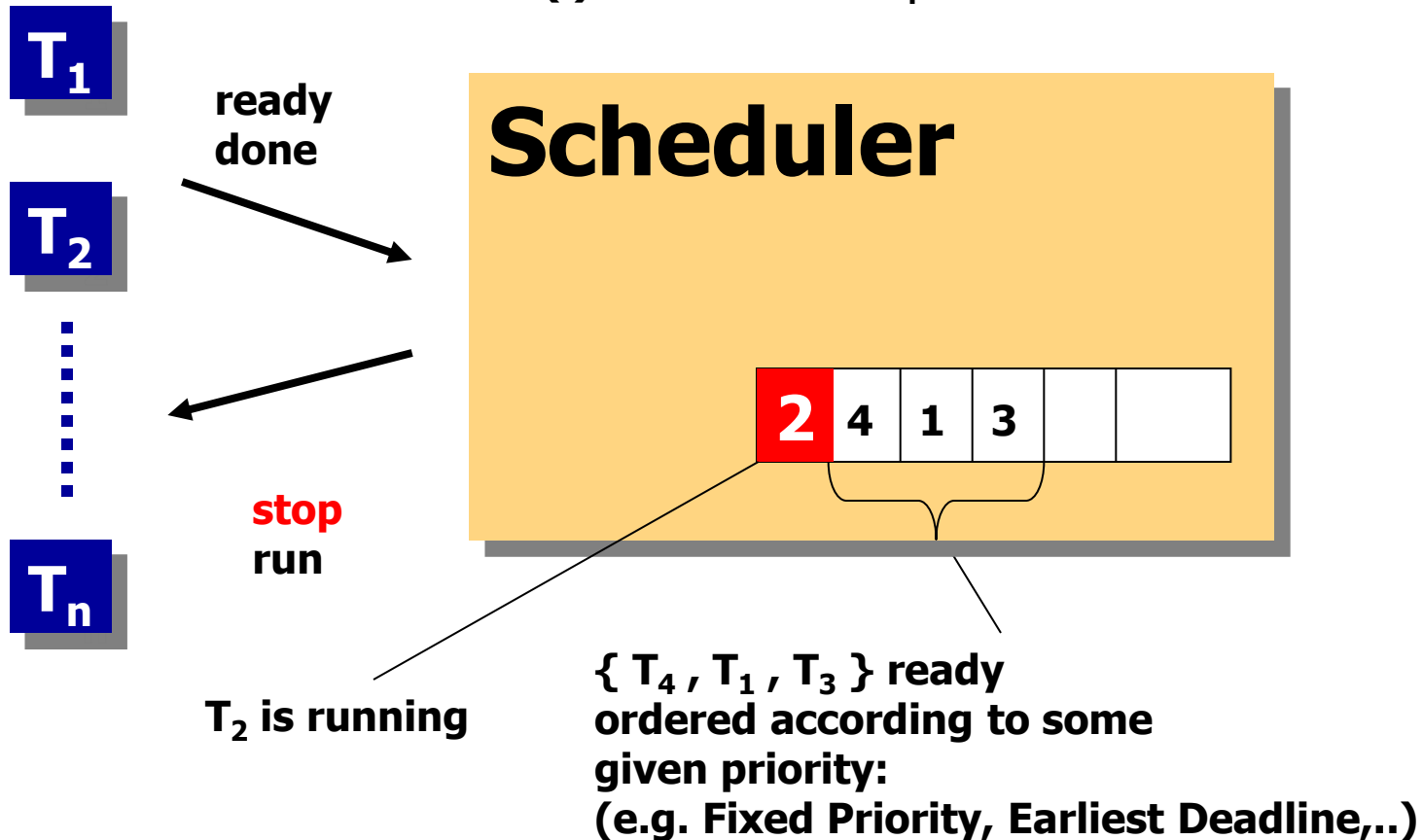
Schedulability & Performance Analysis



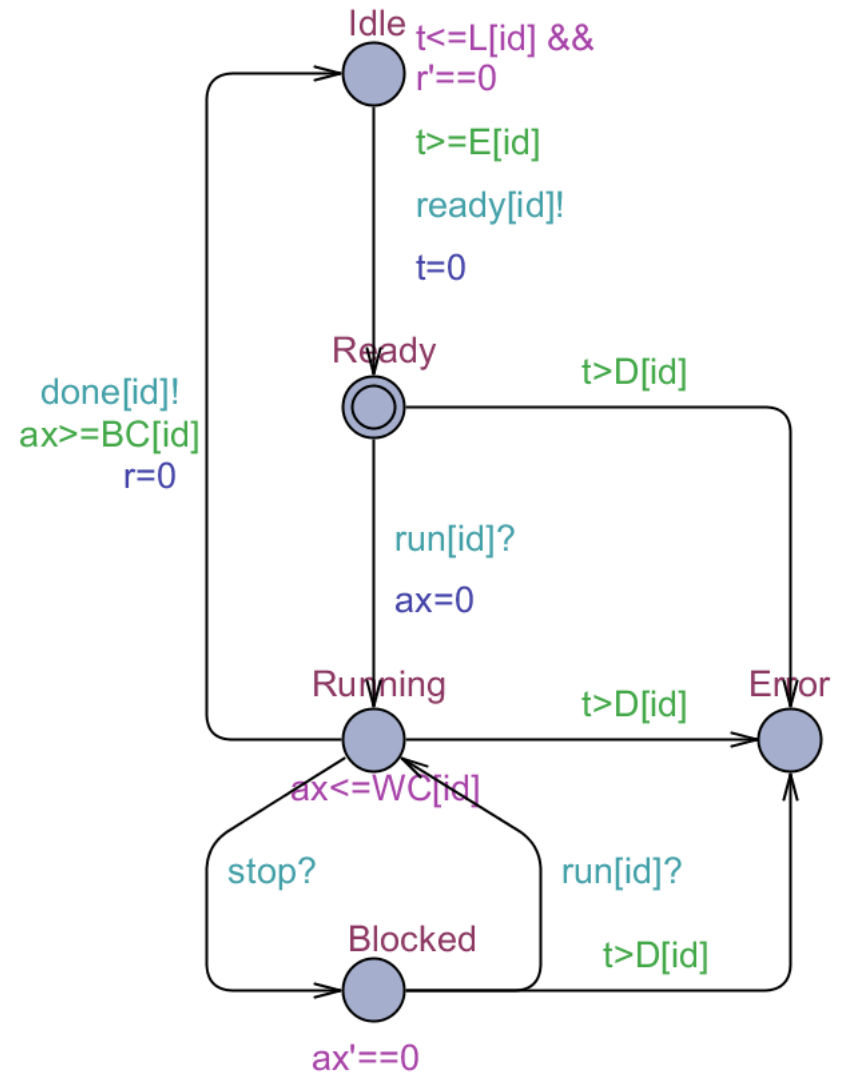
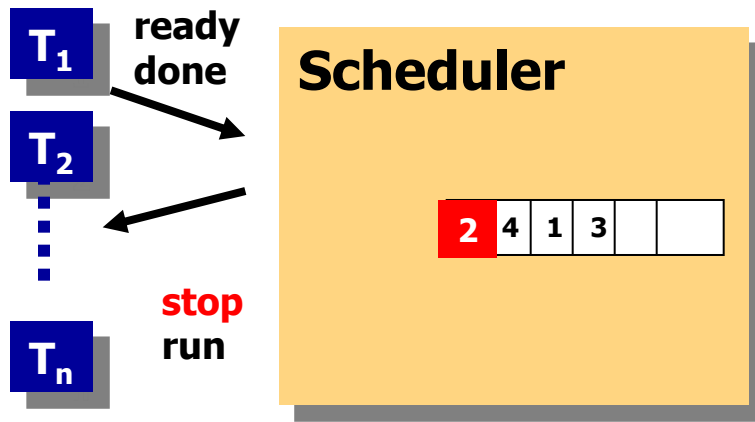
Task Scheduling

utilization of CPU

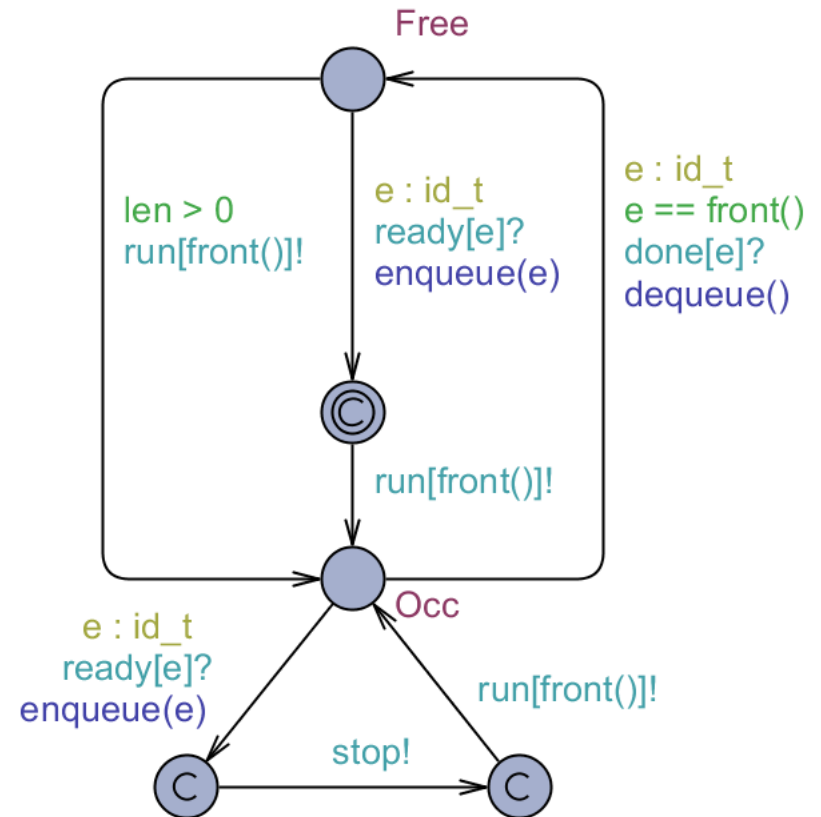
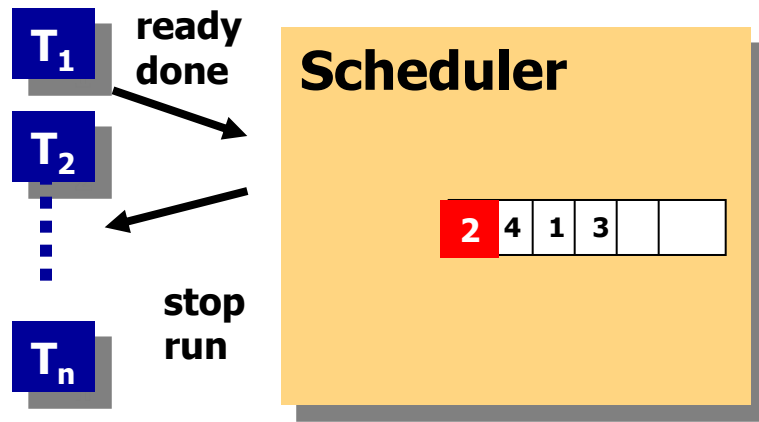
$P(i)$, **UNI**[$E(i)$, $L(i)$], .. : period or
earliest/latest arrival or .. for T_i
 $C(i)$, **UNI**[$BC(i)$, $WC(i)$] : execution time for T_i
 $D(i)$: deadline for T_i



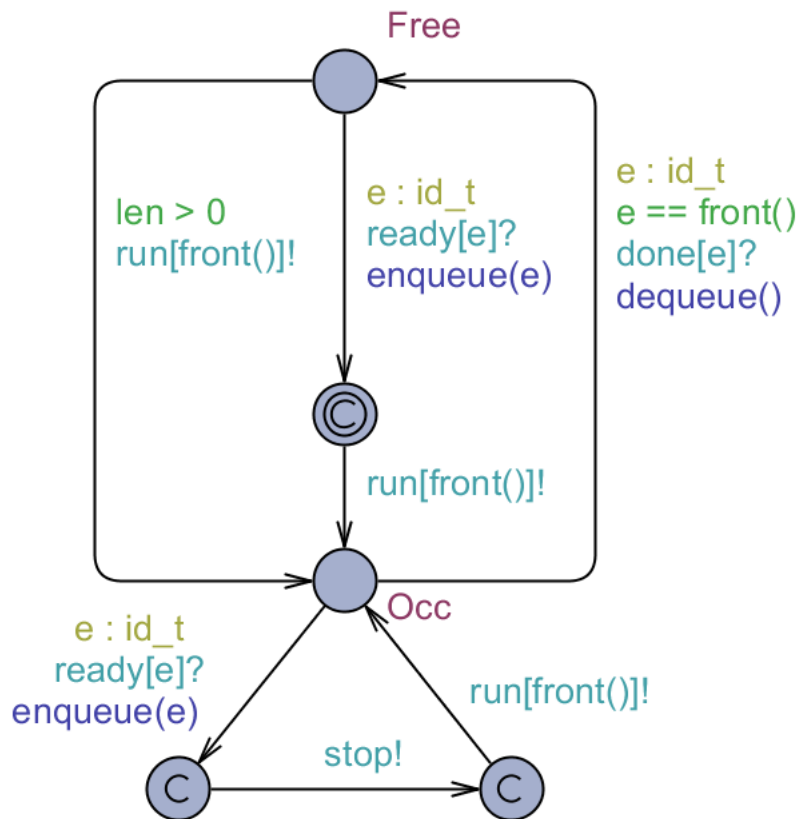
Modeling Task



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--;
    }
  }
}

```

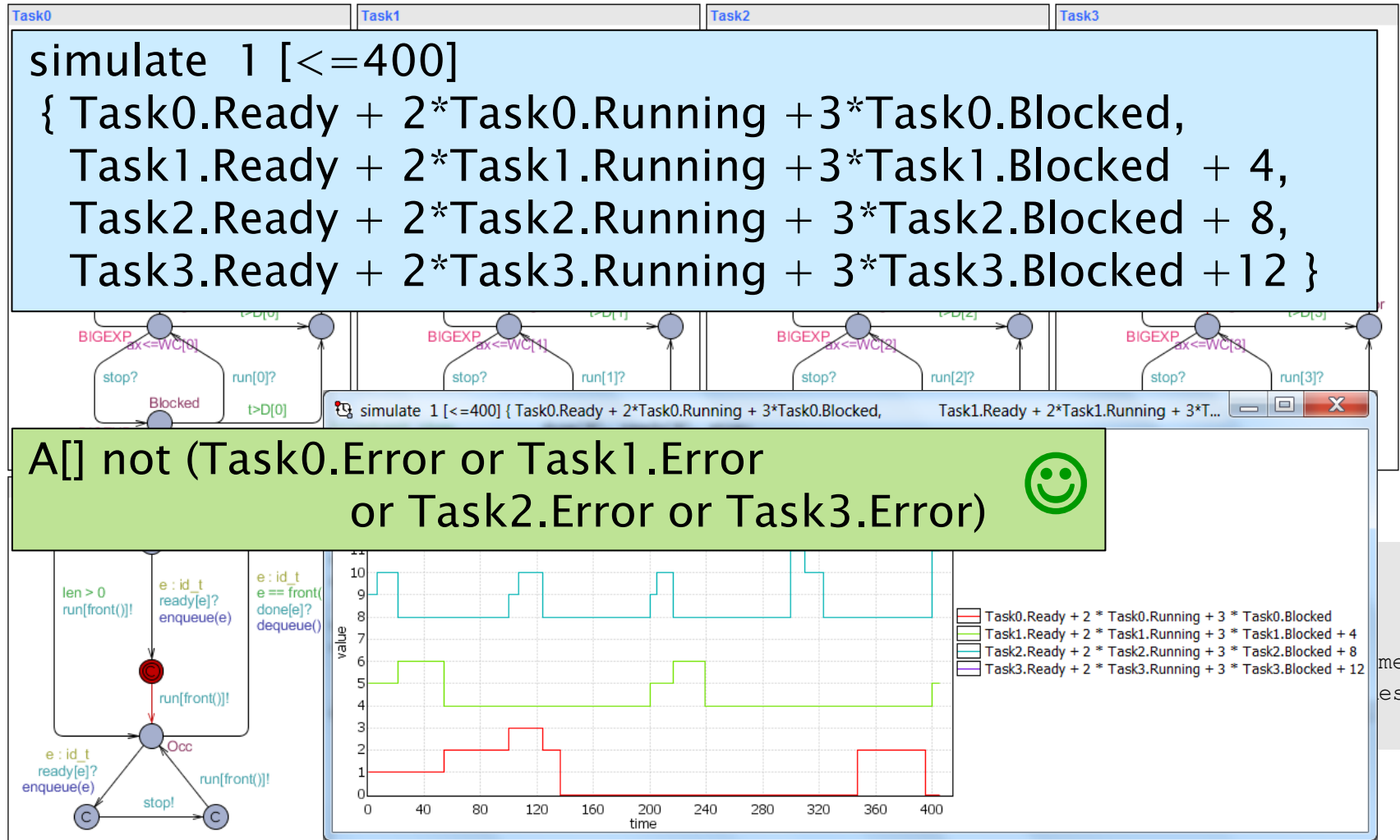
```

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

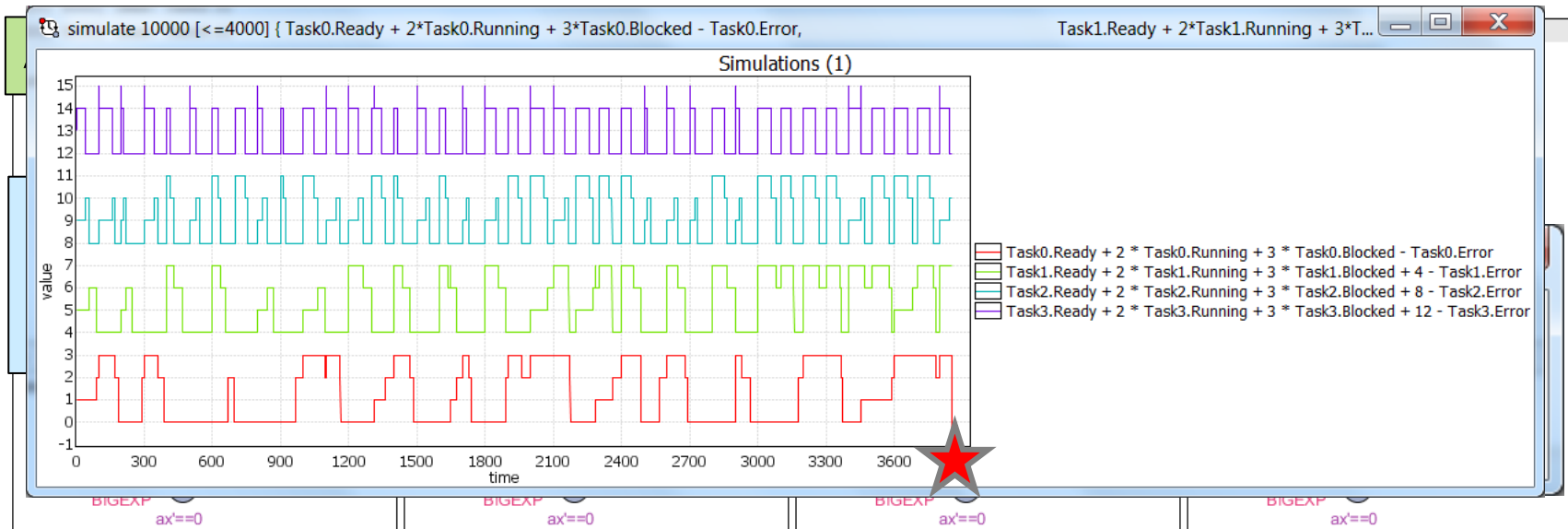
```



Schedulability Analysis

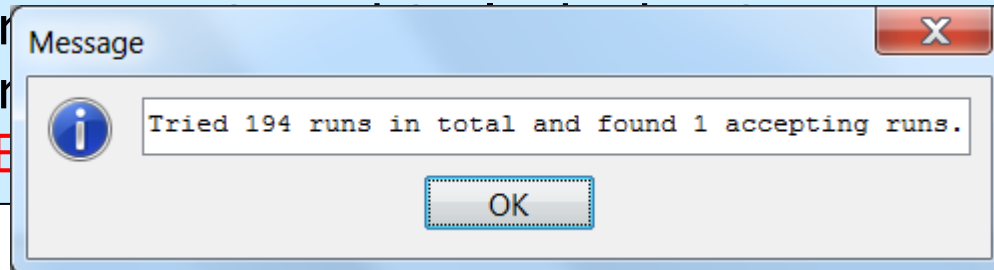


Schedulability Analysis

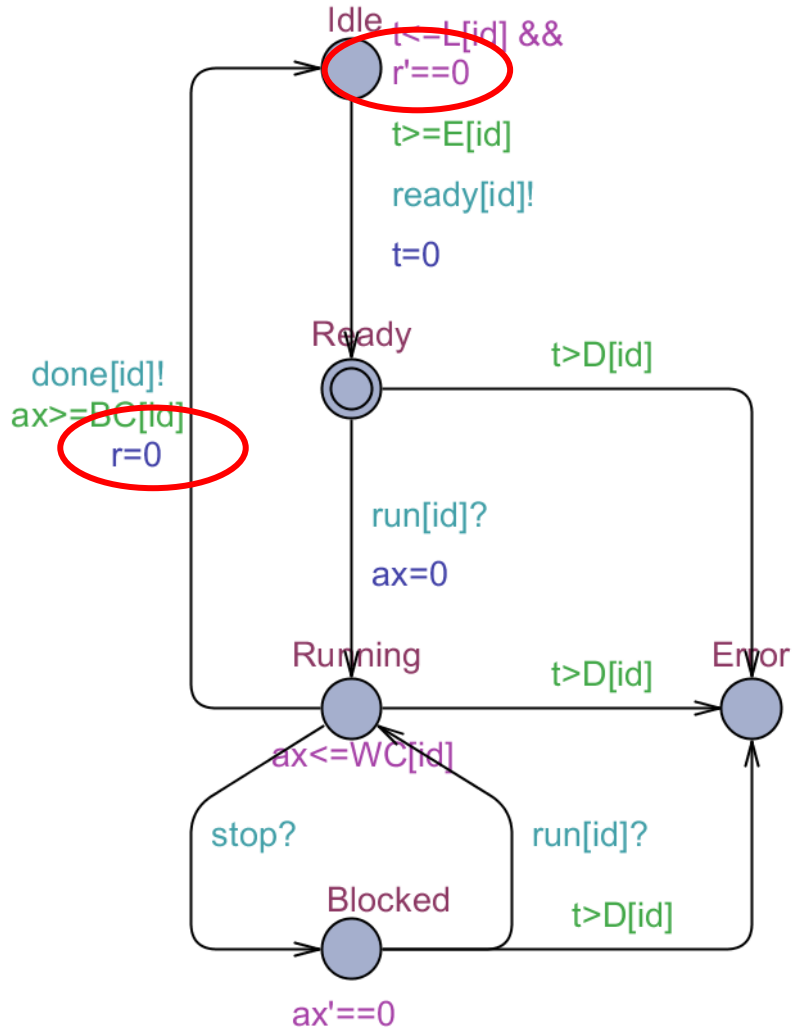


```
simulate 10000 [<=400]
```

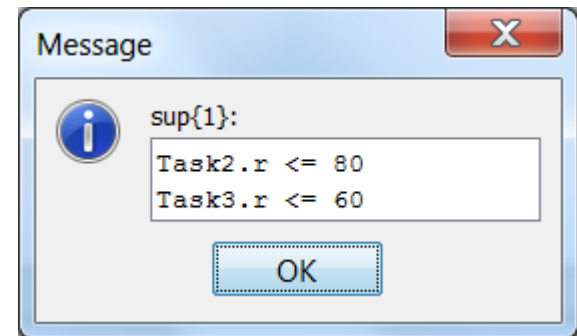
```
{ Task0.Ready + 2*Task0.Running + 3*Task0.Blocked,  
  Task1.Ready + 2*Task1.Running + 3*Task1.Blocked + 4,  
  Task2.Ready + 2*Task2.Running + 3*Task2.Blocked + 8,  
  Task3.Ready + 2*Task3.Running + 3*Task3.Blocked + 12 - Task3.Error  
: 1 : (Task0.Error or Task1.Error)
```



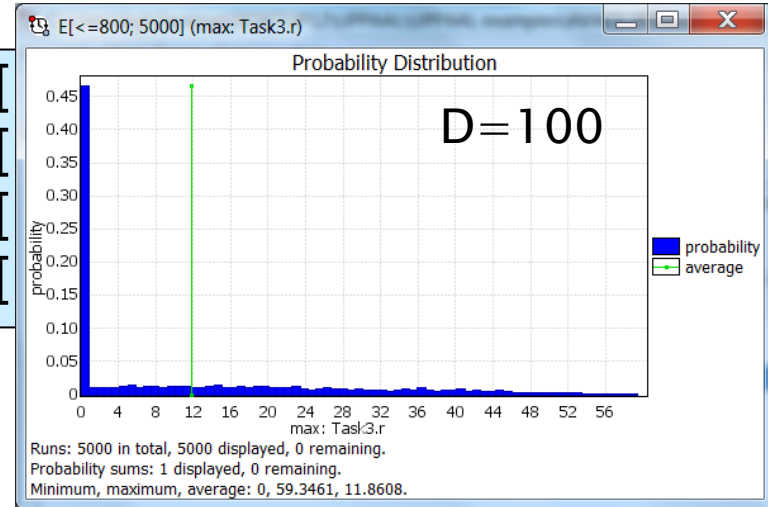
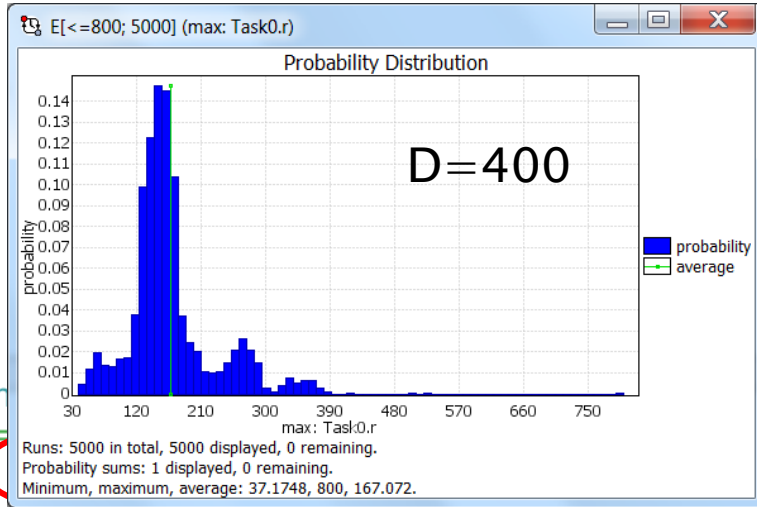
Performance Analysis



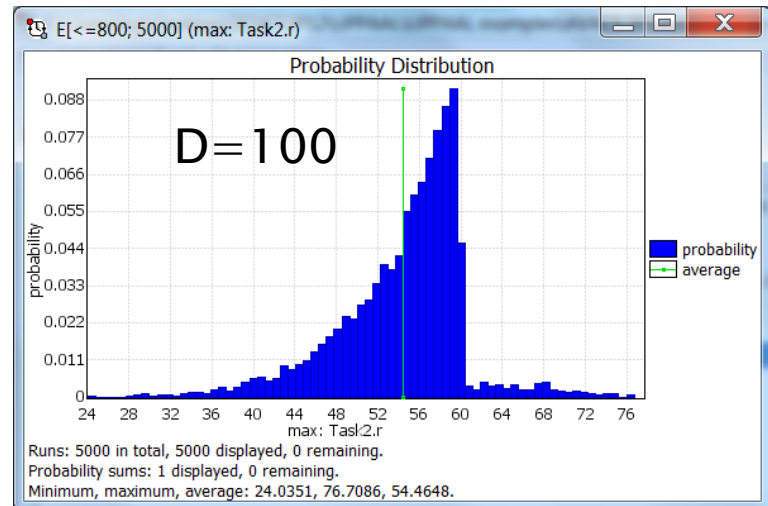
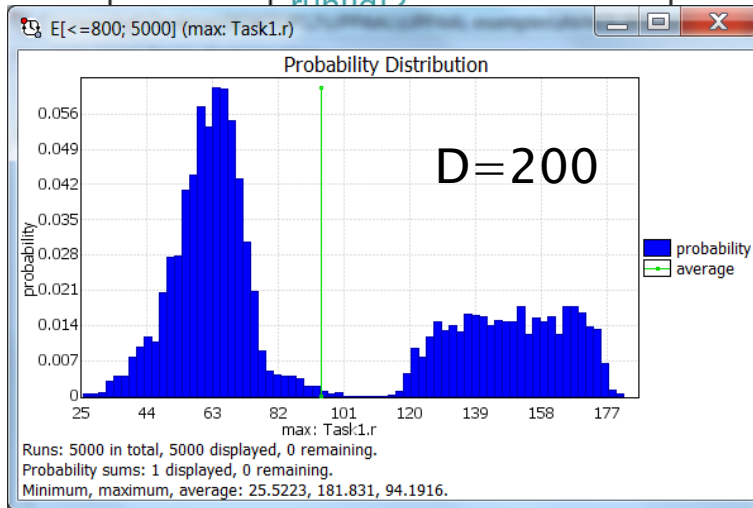
sup : Task2.r, Task3.r



Performance Analysis



E[
E[
E[
E[



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, the 'Transition chooser' shows a list of transitions with 'Scheduler' selected. Below it, a 'Delay' field is set to 13.5, and 'Trace controls' include buttons for 'First', 'Prev', 'Play', and 'Next'. A 'Speeder' slider is positioned between 'Slow' and 'Fast'. The 'Simulation Trace' window 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 central 'Drag out' area contains a list of task queues: 'cycleCount = 0', 'ctask = 7', 'taskqueue[0] = 8', 'taskqueue[1] = 9', 'taskqueue[2] = 10', 'taskqueue[3] = 11', 'taskqueue[4] = 12', 'taskqueue[5] = 13', 'taskqueue[6] = 14', 'taskqueue[7] = 16', 'taskqueue[8] = 17', 'taskqueue[9] = 23', 'taskqueue[10] = 24', 'taskqueue[11] = 25', 'taskqueue[12] = 26', 'taskqueue[13] = 27', 'taskqueue[14] = 28', 'taskqueue[15] = 29', 'taskqueue[16] = 30', 'taskqueue[17] = 33', 'taskqueue[18] = 0', 'taskqueue[19] = 0', 'taskqueue[20] = 0', 'taskqueue[21] = 0', 'taskqueue[22] = 0', 'taskqueue[23] = 0', 'taskqueue[24] = 0', 'taskqueue[25] = 0', 'taskqueue[26] = 0', 'taskqueue[27] = 0', 'taskqueue[28] = 0', 'taskqueue[29] = 0', 'taskqueue[30] = 0', 'taskqueue[31] = 0', 'taskqueue[32] = 0', 'taskqueue[33] = 0', 'running[0] = 0', 'running[1] = 0', 'running[2] = 0'. Three state transition diagrams are shown: 'Scheduler' with states 'initialize/main', 'Running', 'Preempt', and 'Schedule'; 'Bkgnd_P' with states 'starting', 'Idle', 'Ready', and 'Error'; and 'secondF_2' with states 'Idle', 'Blocked', 'WaitForCPU', 'WaitForOther', and 'Handle Pending TCWithBoth'. The 'secondF_1' diagram is also visible, showing states like 'Blocked', 'Reschedule', 'WaitForCPU', and 'DetermineUnit HealthWithSgm_R'.



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:

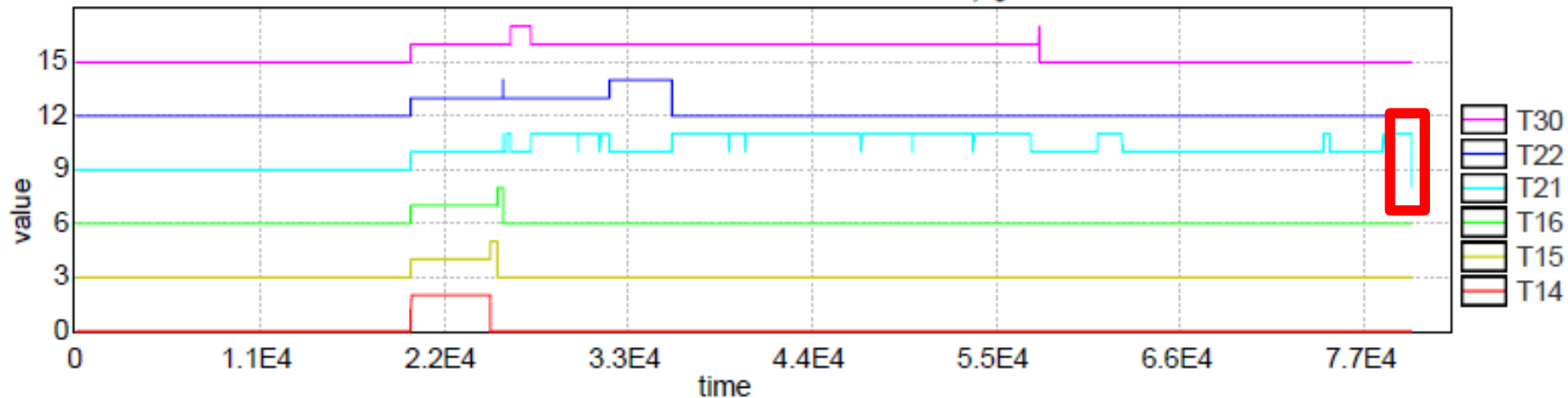
- can show *presence of errors* but not absence.
- \Rightarrow suitable for *disproving schedulability*.

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



SMC Simulation to Find Error

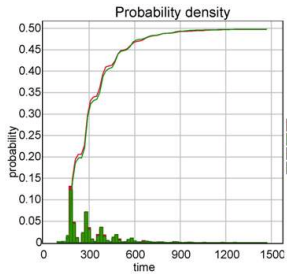
Herschel deadline violation with $f = 50\%$:



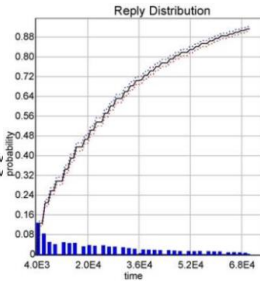
```
simulate 10000 [ <= 300 ] {  
  (T(1).Ready + T(1).Computing + T(1).Release + runs[1] - 2 * T(1)  
  (T(2).Ready + T(2).Computing + T(2).Release + runs[2] - 2 * T(1)  
  (T(3).Ready + T(3).Computing + T(3).Release + runs[3] - 2 * T(1)  
} : 1 : error
```



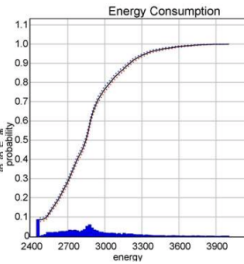
Other Case Studies



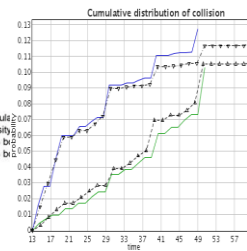
FIREWIRE



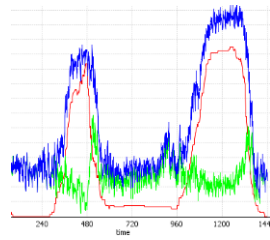
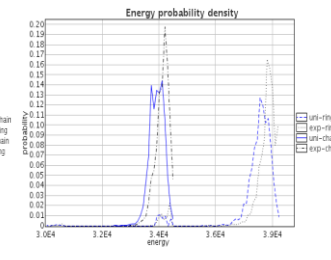
BLUETOOTH



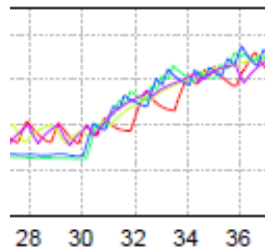
10 node LMAC



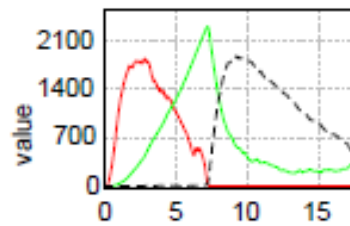
Schedulability Analysis for Mix Cr Sys



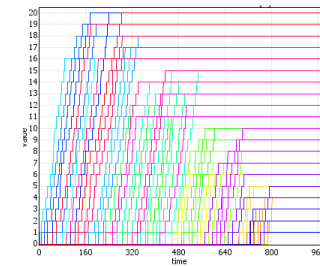
Smart Grid Demand / Response



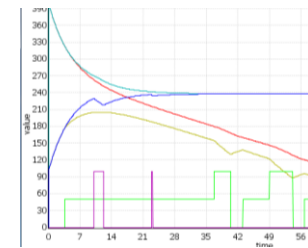
Energy Aware Buildings



Genetic Oscillator (HBS)



Passenger Seating in Aircraft



Battery Scheduling (SENSATION) Erik Wogensen



Formal & Informal Methods

- Model Checking vs Stat MC, Simulation
- Qualitative vs Quantitative (metrics)
- State Space Expl vs Confidence Expl
- Correctness (overap) vs Counterex (underap)
- Worst Case vs Expected Case
- Synthesis on abstract models vs Performance eval on refined models
-



www.uppaal.org



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 [≤ 100] { e1, e2, x1 }`
- `simulate 100 [≤ 10] { e } : 2 : goal`

Exercise 28 (Jobshop scheduling part 2)

