Decidability and Symbolic Verification

Kim G. Larsen
Aalborg University, DENMARK
Decidability
Reachability?

Reachable from initial state (L0, x=0, y=0) ?

OBSTACLE: Uncountably infinite state space

\[ L \times \mathbb{R}^C \]
locations clock-valuations

Kim Larsen [3]
The Region Abstraction

Region $R$ defined by:

\[
\begin{cases}
0 < x < 1 \\
0 < y < 1 \\
y < x
\end{cases}
\]

- “compatibility” between regions and constraints
- “compatibility” between regions and time elapsing

\[\leadsto\] an equivalence of finite index

a time-abstract bisimulation
Time Abstracted Bisimulation

This is a relation between $\bullet$ and $\bullet$ such that:

$\forall a \bullet \Rightarrow \exists a \bullet$

$\forall d > 0 \bullet \Rightarrow \exists d' > 0 \bullet$

$\delta(d) \bullet \Rightarrow \delta(d') \bullet$

... and vice-versa (swap $\bullet$ and $\bullet$).
Regions – From Infinite to Finite

THM [AD90]
Reachability is decidable (and PSPACE-complete) for timed automata

THM [CY90]
Time-optimal reachability is decidable (and PSPACE-complete) for timed automata
Region Graph

It “mimicks” the behaviours of the clocks.
Region Automaton = Finite Bisimulation Quotient

\[ L(\text{reg. aut.}) = \text{UNTIME}(L(\text{timed aut.})) \]
An Example

[AD94]
**LARGE**: exponential in the number of clocks and in the constants (if encoded in binary). The number of regions is

\[ \prod_{x \in X} (2M_x + 2) \cdot |X| \cdot 2^{|X|} \]
Fundamental Results

- Reachability
- Model-checking
  - TCTL \(\text{PSPACE-c}\); MTL \(\text{UNDICIDABLE}\); MITL \(\text{PSPACE-c}\)
- Bisimulation, Simulation
  - Timed \(\text{EXPTIME-c}\); Untimed
- Trace-inclusion
  - Timed \(\text{UNDICIDABLE}\); Untimed \(\text{EXPSPACE-c}\)
Symbolic Verification

The UPPAAL Verification Engine
Regions – From Infinite to Finite

Theorem

The number of regions is \( n! \cdot 2^n \cdot \prod_{x \in C}(2c_x + 2). \)

Region construction: [AD94]
In practice: Zones

Kim Larsen [13]
A zone $Z$:

\[
1 \leq x \leq 2 \quad \land \\
0 \leq y \leq 2 \quad \land \\
x - y \geq 0
\]
Zones – Operations

\[(n, \ 2 \leq x \leq 4 \land 1 \leq y \leq 3 \land y-x \leq 0 )\]

\[(n, \ 2 \leq x \land 1 \leq y \land -3 \leq y-x \leq 0 )\]

\[(n, \ 2 \leq x \land 1 \leq y \leq 3 \land y-x \leq 0 )\]

\[(n, \ x=0 \land 1 \leq y \leq 3 )\]

\[(n, \ 2 \leq x \leq 4 \land 1 \leq y )\]

\[(n, \ 2 \leq x \land 1 \leq y \land -3 \leq y-x \leq 0 )\]

\[(n, \ 2 \leq x \land 1 \leq y \leq 3 \land y-x \leq 0 )\]

\[\text{Delay}\]

\[\text{Delay (stopwatch)}\]

\[\text{Convex Hull}\]

\[\text{Extrapolation}\]

\[\text{Reset}\]
Symbolic Transitions

\[(\ell, 1 \leq x \leq 4, 1 \leq y \leq 3) \Rightarrow_a (\ell', 3 < x, y = 0)\]
Forward Reachability

INITIAL
\[ \text{Passed} := \emptyset; \]
\[ \text{Waiting} := \{(n_0, Z_0)\} \]

REPEAT
pick \((n, Z)\) in Waiting
if \((n, Z) = \text{Final}\) return true
for all \((n, Z) \rightarrow (n', Z'):\)
if for some \((n', Z'')\) \(Z' \subseteq Z''\) continue
else add \((n', Z')\) to Waiting
move \((n, Z)\) to Passed

UNTIL \text{Waiting} = \emptyset
return false

---


Kim Larsen [17]
Forward Reachability

\[ \text{INITIAL } \text{Passed} := \emptyset; \]
\[ \text{Waiting} := \{(n_0, Z_0)\} \]

\[ \text{REPEAT} \]
\[ \text{pick } (n, Z) \text{ in Waiting} \]
\[ \text{if } (n, Z) = \text{Final return true} \]
\[ \text{if for all } (n, Z) \rightarrow (n', Z'):\]
\[ \text{if for some } (n', Z'') \text{ } Z' \subseteq Z'' \text{ continue} \]
\[ \text{else add } (n', Z') \text{ to Waiting} \]
\[ \text{move } (n, Z) \text{ to Passed} \]

\[ \text{UNTIL } \text{Waiting} = \emptyset \]
\[ \text{return false} \]
Forward Reachability

**Initial** -> **Final**?

**Initial** - > **Final ?**

**INITIAL**  
Passed := Ø;  
Waiting := {(n₀,Z₀)}

**REPEAT**

pick (n,Z) in Waiting
if (n,Z) = Final return true
for all (n,Z)→(n',Z'):
  if for some (n',Z'') Z' ⊆ Z'' continue
else add (n',Z') to Waiting
move (n,Z) to Passed

**UNTIL** Waiting = Ø
return false

Forward Reachability

INITIAL  Passed := ∅;
          Waiting := {(n_0,Z_0)}

REPEAT
  pick (n,Z) in Waiting
  if (n,Z) = Final return true
  for all (n,Z)→(n',Z'):
    if for some (n',Z'') Z'⊆ Z'' continue
    else add (n',Z') to Waiting
  move (n,Z) to Passed
UNTIL  Waiting = ∅
return false
Forward Reachability

INITIAL
Passed := Ø;
Waiting := { (n₀, Z₀) }

REPEAT
pick (n, Z) in Waiting
if (n, Z) = Final return true
for all (n, Z) → (n', Z'):
  if for some (n', Z'') Z' ⊆ Z'' continue
else add (n', Z') to Waiting
move (n, Z) to Passed

UNTIL Waiting = Ø
return false
**Forward Reachability**

**INITIAL**
- Passed := Ø;
- Waiting := \{(n_0,Z_0)\}

**REPEAT**
- pick \((n,Z)\) in Waiting
- if \((n,Z) = \text{Final}\) return true
- for all \((n,Z) \rightarrow (n',Z')\):
  - if for some \((n',Z'')\) \(Z' \subseteq Z''\) continue
  - else add \((n',Z')\) to Waiting
- move \((n,Z)\) to Passed

**UNTIL**
- Waiting = Ø
return false

---

**Verification Theory, Systems and Applications Summer School. September 2013.**

Kim Larsen [22]
INITIAL \hspace{1em} \text{Passed} := \emptyset; \\
\text{Waiting} := \{(n_0, Z_0)\}

REPEAT
\begin{align*}
\text{pick} \ (n, Z) \ \text{in} \ \text{Waiting} \\
\text{if} \ (n, Z) = \text{Final} \ \text{return true} \\
\text{for all} \ (n, Z) \rightarrow (n', Z'):
\begin{align*}
\text{if for some} \ (n', Z'') \ Z' \subseteq Z'' \ \text{continue} \\
\text{else add} \ (n', Z') \ \text{to} \ \text{Waiting} \\
\text{move} \ (n, Z) \ \text{to} \ \text{Passed}
\end{align*}
\end{align*}

UNTIL \ \text{Waiting} = \emptyset \\
\text{return false}
Symbolic Exploration

Reachability?

L0

x := 0

y := 0

y <= 2

x <= 2

y <= 2, x >= 4

L1

Symbolic Exploration

Reachable?
Symbolic Exploration

Reachable?

Kim Larsen [27]
Symbolic Exploration

Reachable?

Symbolic Exploration

Reachable?


Kim Larsen [30]
Symbolic Exploration

Reachable?

Kim Larsen [31]
Symbolic Exploration


Kim Larsen [32]
Datastructures for Zones

- Difference Bounded Matrices (DBMs)
- Minimal Constraint Form
  [RTSS97]
- Clock Difference Diagrams
  [CAV99]
Inclusion Checking (DBMs)

Inclusion

D1
\[
\begin{align*}
x & \leq 1 \\
y - x & \leq 2 \\
z - y & \leq 2 \\
z & \leq 9
\end{align*}
\]

Graph

D2
\[
\begin{align*}
x & \leq 2 \\
y - x & \leq 3 \\
y & \leq 3 \\
z - y & \leq 3 \\
z & \leq 7
\end{align*}
\]

Graph

Bellman 1958, Dill 1989


Kim Larsen [34]
Future (DBMs)

\[ 1 \leq x \leq 4 \]
\[ 1 \leq y \leq 3 \]

Shortest Path Closure

Remove upper bounds on clocks

\[ 1 \leq x, 1 \leq y \]
\[ -2 \leq x-y \leq 3 \]
Reset (DBMs)

\begin{align*}
1 \leq x, &
1 \leq y \\
-2 \leq x-y &
\leq 3
\end{align*}

Remove all bounds involving \( y \) and set \( y \) to 0.

\begin{align*}
\{y\}D
\quad &
\quad y=0, 1 \leq x
\end{align*}
Difference Bounded Matrices

\[
\begin{align*}
x_1 - x_2 & \leq 4 \\
x_2 - x_1 & \leq 10 \\
x_3 - x_1 & \leq 2 \\
x_2 - x_3 & \leq 2 \\
x_0 - x_1 & \leq 3 \\
x_3 - x_0 & \leq 5
\end{align*}
\]

Shortest Path Closure
\[O(n^3)\]
Minimal Constraint Form

\[
\begin{align*}
x_1 - x_2 &\leq 4 \\
x_2 - x_1 &\leq 10 \\
x_3 - x_1 &\leq 2 \\
x_2 - x_3 &\leq 2 \\
x_0 - x_1 &\leq 3 \\
x_3 - x_0 &\leq 5
\end{align*}
\]

Graph 1: Shortest Path Closure \(O(n^3)\)

Graph 2: Shortest Path Reduction \(O(n^3)\)

Graph 3: Space worst \(O(n^2)\) practice \(O(n)\)

RTSS 1997


Kim Larsen [38]
Earlier Termination

\[
\text{INITIAL} \quad \text{Passed} := \emptyset; \\
\text{Waiting} := \{(n_0, Z_0)\}
\]

\[
\text{REPEAT} \\
\quad \text{pick } (n, Z) \text{ in Waiting} \\
\quad \text{if } (n, Z) = \text{Final return true} \\
\quad \text{for all } (n, Z) \rightarrow (n', Z') : \\
\quad \quad \text{if for some } (n', Z')' \quad Z' \subseteq Z'' \quad \text{continue} \\
\quad \quad \text{else add } (n', Z') \text{ to Waiting} \\
\quad \text{move } (n, Z) \text{ to Passed}
\]

\[
\text{UNTIL} \quad \text{Waiting} = \emptyset \\
\text{return false}
\]
Earlier Termination

\[
\begin{align*}
\text{INITIAL} & \quad \text{Passed} := \emptyset; \\
& \quad \text{Waiting} := \{(n_0, Z_0)\}
\end{align*}
\]

\[
\text{REPEAT}
\begin{align*}
\text{pick} & \quad (n, Z) \text{ in Waiting} \\
\text{if} & \quad (n, Z) = \text{Final} \quad \text{return true} \\
\text{for all} & \quad (n, Z) \rightarrow (n', Z'):\n\quad \text{if for some} \quad (n', Z'') \quad Z' \subseteq Z'' \quad \text{continue} \\
\quad \text{else} & \quad \text{add} \quad (n', Z') \text{ to Waiting} \\
\text{move} & \quad (n, Z) \text{ to Passed}
\end{align*}
\]

UNTIL \quad \text{Waiting} = \emptyset

\text{return false}

Init -> Final ?
Nodes labeled with differences
Maximal sharing of substructures (also across different CDDs)
Maximal intervals
Linear–time algorithms for set–theoretic operations.

NDD’s Maler et. al
DDD’s Møller, Lichtenberg
Verification Options
Verification Options

Search Order
- Depth First
- Breadth First

State Space Reduction
- None
- Conservative
- Aggressive

State Space Representation
- DBM
- Compact Form
- Under Approximation
- Over Approximation

Diagnostic Trace
- Some
- Shortest
- Fastest

Extrapolation
- Hash Table size
- Reuse
State Space Reduction

Cycles:
Only symbolic states involving loop-entry points need to be saved on Passed list
To Store or Not To Store

![Diagram of an audio protocol with 117 total states, 81 entrypoint states, and 9 states with time overhead less than 10%.]

Behrmann, Larsen, Pelanek 2003


Kim Larsen [45]
Question:

G ∈ R ?

How to use:

G ∈ O ?
G ∈ U ?

G ∈ U  ⇒  G ∈ R
¬(G ∈ O)  ⇒  ¬(G ∈ R)
Over-approximation Convex Hull

TACAS04: An EXACT method performing as well as Convex Hull has been developed based on abstractions taking max constants into account distinguishing between clocks, locations and $\leq \& \geq$. 
Under-approximation
Bitstate Hashing
Under-approximation
Bitstate Hashing

Passed

Waiting

Final

Bitarray

Hashfunction $F$

Pass$=\text{Bitarray}$

UPPAAL
4 - 512 Mbits


Kim Larsen [49]
Extrapolation


Kim Larsen [50]
Forward Symbolic Exploration

\[ y := 0, \]
\[ x := 0 \]

\[ x \geq 1 \land y = 1, \]
\[ y := 0 \]

\( (y \leq 1) \)

TERMINATION not guaranteed

Need for Finite Abstractions
Abstractions

\[ a : \mathcal{P}(R^{X}_{\geq 0}) \leftrightarrow \mathcal{P}(R^{X}_{\geq 0}) \text{ such that } W \subseteq a(W) \]

\[
\begin{align*}
(\ell, W) & \Rightarrow (\ell', W') \\
(\ell, W) & \Rightarrow_{a} (\ell', a(W'))
\end{align*}
\]

if \( W = a(W) \)

We want \( \Rightarrow_{a} \) to be:

- sound & complete wrt reachability
- finite
- easy to compute
- as coarse as possible
Abstraction by Extrapolation

Let $k$ be the largest constant appearing in the TA

\[
\begin{align*}
&x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \\
&x_1 \rightarrow \ldots \rightarrow x_2 \rightarrow \ldots \rightarrow x_3 \rightarrow x_0
\end{align*}
\]

Sound & Complete Ensures Termination
Will generate all symbolic states of the form

$$(l_2, x \in [0,14], y \in [5,14n], y-x \in [5,14n-14])$$

for $n \leq 10^6/14$ !!

But $y \geq 10^6$ is not RELEVANT in $l_2$
Location Dependent Constants

\[
\begin{align*}
          k_x = 5 & \quad k_y = 10^6 \\
\end{align*}
\]

\[k_x^i = 5 \quad \text{for } i \in \{1, 2, 3\}\]

\[k_y^4 = 10^6\]

\[k_j^i\] may be found as solution to simple linear constraints!

Active Clock Reduction:

\[k_j^i = -\infty\]
<table>
<thead>
<tr>
<th></th>
<th>Constant BIG</th>
<th>Global Method</th>
<th>Active-clock Reduction</th>
<th>Local Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Naive Example</strong></td>
<td>$10^3$</td>
<td>0.05s/1MB</td>
<td>0.05s/1MB</td>
<td>0.00s/1MB</td>
</tr>
<tr>
<td></td>
<td>$10^4$</td>
<td>4.78s/3MB</td>
<td>4.83s/3MB</td>
<td>0.00s/1MB</td>
</tr>
<tr>
<td></td>
<td>$10^5$</td>
<td>484s/13MB</td>
<td>480s/13MB</td>
<td>0.00s/1MB</td>
</tr>
<tr>
<td></td>
<td>$10^6$</td>
<td>stopped</td>
<td>stopped</td>
<td>0.00s/1MB</td>
</tr>
<tr>
<td><strong>Two Processes</strong></td>
<td>$10^3$</td>
<td>3.24s/3MB</td>
<td>3.26s/3MB</td>
<td>0.01s/1MB</td>
</tr>
<tr>
<td></td>
<td>$10^4$</td>
<td>5981s/9MB</td>
<td>5978s/9MB</td>
<td>0.37s/2MB</td>
</tr>
<tr>
<td></td>
<td>$10^5$</td>
<td>stopped</td>
<td>stopped</td>
<td>72s/5MB</td>
</tr>
<tr>
<td><strong>Asymmetric Fischer</strong></td>
<td>$10^3$</td>
<td>0.01s/1MB</td>
<td>0.01s/1MB</td>
<td>0.01s/1MB</td>
</tr>
<tr>
<td></td>
<td>$10^4$</td>
<td>2.20s/3MB</td>
<td>2.20s/3MB</td>
<td>0.85s/2MB</td>
</tr>
<tr>
<td></td>
<td>$10^5$</td>
<td>333s/19MB</td>
<td>333s/19MB</td>
<td>160s/13MB</td>
</tr>
<tr>
<td></td>
<td>$10^6$</td>
<td>33307s/122MB</td>
<td>33238s/122MB</td>
<td>16330s/65MB</td>
</tr>
<tr>
<td><strong>Bang &amp; Olufsen</strong></td>
<td>25000</td>
<td>stopped</td>
<td>159s/243MB</td>
<td>123s/204MB</td>
</tr>
</tbody>
</table>
Lower and Upper Bounds

\[ \text{Given that } x \leq 10^6 \text{ is an upper bound implies that} \]

\[(l, v_x, v_y) \text{ simulates } (l, v'_x, v_y) \]

\[\text{whenever } v'_x \geq v_x \geq 10.\]

For reachability downward closure wrt simulation suffices!
## Advanced Extrapolation

### Fischer

<table>
<thead>
<tr>
<th>Model</th>
<th>Classical</th>
<th>Loc. dep. Max</th>
<th>Loc. dep. LU</th>
<th>Convex Hull</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-n1</td>
<td>-n2</td>
<td>-n3</td>
<td>-A</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>States</td>
<td>Mem</td>
<td>Time</td>
</tr>
<tr>
<td>f5</td>
<td>4.02</td>
<td>82,685</td>
<td>5</td>
<td>0.24</td>
</tr>
<tr>
<td>f6</td>
<td>597.04</td>
<td>1,489,230</td>
<td>49</td>
<td>6.67</td>
</tr>
<tr>
<td>f7</td>
<td>352.67</td>
<td>1,620,542</td>
<td>46</td>
<td>0.47</td>
</tr>
<tr>
<td>f8</td>
<td>2.11</td>
<td>164,528</td>
<td>6</td>
<td>9.11</td>
</tr>
<tr>
<td>f9</td>
<td>8.76</td>
<td>598,662</td>
<td>19</td>
<td>208,744</td>
</tr>
<tr>
<td>f10</td>
<td>37.26</td>
<td>2,136,980</td>
<td>68</td>
<td>39.13</td>
</tr>
<tr>
<td>f11</td>
<td>152.44</td>
<td>7,510,382</td>
<td>268</td>
<td></td>
</tr>
</tbody>
</table>

### CSMA/CD

<table>
<thead>
<tr>
<th>Model</th>
<th>Classical</th>
<th>Loc. dep. Max</th>
<th>Loc. dep. LU</th>
<th>Convex Hull</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-n1</td>
<td>-n2</td>
<td>-n3</td>
<td>-A</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>States</td>
<td>Mem</td>
<td>Time</td>
</tr>
<tr>
<td>c5</td>
<td>0.55</td>
<td>27,174</td>
<td>3</td>
<td>0.14</td>
</tr>
<tr>
<td>c6</td>
<td>19.39</td>
<td>287,109</td>
<td>11</td>
<td>3.63</td>
</tr>
<tr>
<td>c7</td>
<td>195.35</td>
<td>813,924</td>
<td>29</td>
<td>0.28</td>
</tr>
<tr>
<td>c8</td>
<td>0.98</td>
<td>50,058</td>
<td>5</td>
<td>0.98</td>
</tr>
<tr>
<td>c9</td>
<td>2.90</td>
<td>132,623</td>
<td>12</td>
<td>2.90</td>
</tr>
<tr>
<td>c10</td>
<td>8.42</td>
<td>341,452</td>
<td>29</td>
<td>5.48</td>
</tr>
<tr>
<td>c11</td>
<td>24.13</td>
<td>859,265</td>
<td>76</td>
<td>15.66</td>
</tr>
<tr>
<td>c12</td>
<td>68.20</td>
<td>2,122,286</td>
<td>202</td>
<td>43.10</td>
</tr>
<tr>
<td>bus</td>
<td>102.28</td>
<td>6,727,443</td>
<td>303</td>
<td>66.54</td>
</tr>
<tr>
<td>philips</td>
<td>0.16</td>
<td>12,823</td>
<td>3</td>
<td>0.09</td>
</tr>
<tr>
<td>sched</td>
<td>17.01</td>
<td>929,726</td>
<td>76</td>
<td>15.09</td>
</tr>
<tr>
<td></td>
<td>55.41</td>
<td>3,636,576</td>
<td>427</td>
<td></td>
</tr>
</tbody>
</table>
Additional “secrets”

- Sharing among symbolic states
  - location vector / discrete values / zones
- Symmetry Reduction
- Sweep Line Method
- Guiding wrt Heuristic Value (CORA)
  - User–supplied / Auto–generated
- “Manual” tricks:
  - active variable reduction
  - Value passing using arrays of channels
Open Problems

- Fully symbolic exploration of TA (both discrete and continuous part)?
- Canonical form for CDD’s?
- Partial Order Reduction?
- Compositional Backwards Reachability?
- Bounded Model Checking for TA?
- Exploitation of multi-core processors?
- ...
Application: Schedulability Analysis
Task Scheduling

utilization of CPU

- $T_2$ is running
- $\{ T_4, T_1, T_3 \}$ ready ordered according to some given priority: (e.g. Fixed Priority, Earliest Deadline, ..)
- $P(i), [E(i), L(i)], .. :$ period or earliest/latest arrival or .. for $T_i$
- $C(i):$ execution time for $T_i$
- $D(i):$ deadline for $T_i$

Scheduler

Kim Larsen [62]
Utilisation-Based Analysis

- A simple sufficient but not necessary schedulability test exists

\[ U \equiv \sum_{i=1}^{N} \frac{C_i}{T_i} \leq N(2^{1/N} - 1) \]

\[ U \leq 0.69 \text{ as } N \rightarrow \infty \]

Where \( C \) is WCET and \( T \) is period

Response Time Equation

\[ R_i = C_i + \sum_{j \in hp(i)} \left[ \frac{R_j}{T_j} \right] C_j \]

Where \( hp(i) \) is the set of tasks with priority higher than task \( i \)

Solve by forming a recurrence relationship:

\[ W_i^{n+1} = C_i + \sum_{j \in hp(i)} \left[ \frac{W_j^n}{T_j} \right] C_j \]

The set of values \( W_i^n, W_i^1, ..., W_i^0 \) is monotonically non-decreasing.

When \( W_i^n = W_i^{n-1} \) the solution to the equation has been found. \( W_i^n \) must not be greater than \( R_i \) (e.g. 0 or \( C_i \))

\( \checkmark \) Simple to perform

- Overly conservative
- Limited settings
- Single-processor

\( \Rightarrow \) Do it in UPPAAL!


Kim Larsen [63]
Modeling Task

Scheduler

Kim Larsen [64]
Modeling Scheduler

Implementation of enqueue/dequeue ⇒ scheduling policy
Modeling Queue

In UPPAAL 4.0
User Defined Function

```
// 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()
`
```

Sort by priority
Schedulability = Safety Property

\[ \neg (\text{Task0.Error} \lor \text{Task1.Error} \lor ...) \]

May be extended with preemption

Kim Larsen [67]
Preemption – Stopwatches!


Defeating undecidability 😊
Stop-Watches

- Make reachability undecidable.
- Over-approximation used in UPPAAL
  - \(\implies\) Safe for positive schedulability results!

- What to do if you violate deadlines?
  - Try to validate the trace using other techniques, e.g., polyhedra.
  - Use SMC!
LAB–Exercises (cont)

www.cs.aau.dk/~kgl/Shanghai2013/exercises

Exercise 1 (Brick Sorter)
Exercise 2 (Coffee Machine)
Exercise 19 (Train Crossing)
Exercise 28 (Jobshop Scheduling)
Exercise 14 (Gossiping Girls)