Checking the consistency of states in case of lost events in a trace

Marie Martin, Michel Dagenais
Agenda

1. Introduction

2. Framework
   - Consistency
   - Certainty
   - Architecture

3. Results
   - Methodology
   - Example
   - View
   - Performance

4. Conclusion
Context

Lost events during tracing
  → Discard & overwrite modes

Parallel analysis of traces
  → No initial global state

What happens to the trace analysis?
An example

**Without missing events**

**With missing events**
Objectives

1. Find incoherent events;
2. Infer information about missing events;
3. Show inconsistency and uncertainty.
Objectives

1. Find incoherent events;
2. Infer information about missing events;
3. Show inconsistency and uncertainty.
Objectives

1. Find incoherent events;
2. Infer information about missing events;
3. Show inconsistency and uncertainty.
Consistency

Definition

Suppose we have read $n$ events from a trace, so that the current state of a state machine $F$ is $S$. The $n+1$ event $e$ is consistent with $F$, if there is a transition from $S$ whose label is $e$, or no transition from any state at all.

So, $e$ is inconsistent if there is a transition in $F$ whose label is $e$, from a state that is not $S$, and there is no possible transition from $S$ with $e$ as its label.
Example

FSM for the process $i$
Example

Without missing events

\[ \text{sched\_switch} \]
\[ \text{[prev\_tid} = i] \]

usermode

wait

\[ \text{sched\_switch} \]
\[ \text{[next\_tid} = i] \]
Example

Without missing events

event is consistent
Example

With missing events

```
sched_switch
[prev_tid = i]

usermode

wait

sched_switch
[next_tid = i]
```
Example

With missing events

```
sched_switch
[prev_tid = i]

usermode

wait

sched_switch
[next_tid = i]
event is inconsistent
```
Certainty

Definition

An event $e$ triggers a **certain** state $S$ if $e$ labels only transitions to $S$.

So, no transition to states other than $S$ are labeled by $e$. 
Example

FSM for the process $i$

FSM for reading in a file

sched_switch
(prev_tid = i AND prev_state = 0)

wait for cpu

sched_switch
(prev_tid = i AND prev_state != 0)

usermode

wait blocked

sched_switch
(next_tid = i)

1 byte
read

2 bytes
read

3 bytes
Example

$sched\_switch$ triggers certain states

FSM for the process $i$

FSM for reading in a file
Example

**FSM for the process $i$**

- **usermode**
  - `$sched\_switch$ [prev\_tid = i AND prev\_state = 0]`
  - `wait\_for\_cpu`

- **wait\_blocked**
  - `$sched\_switch$ [next\_tid = i]`

**FSM for reading in a file**

- 1 byte
  - read
- 2 bytes
  - read
- 3 bytes

`sched\_switch` triggers **certain** states

`read` triggers **uncertain** states
Architecture

**XML state machine**
- Unique attribute
- Non-consuming

**Scenario observer**
- Created if the trace contains 'Lost event'
- Activated when the first 'Lost event' is observed
- Implements algorithm to check the event consistency
Methodology

1. Definition of the FSM in XML
2. Deletion of chosen events from a ’real-world’ trace
3. Execution of the analysis in Trace Compass
4. Comparison of computed inferences with deleted events
Example

Trace with sched_switch events enabled
Deletion of event 777

<table>
<thead>
<tr>
<th>Index</th>
<th>Timestamp</th>
<th>Event type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>775</td>
<td>17:24:27.005</td>
<td>sched_switch</td>
<td>prev_comm=cinnamon, next_comm=swapper/1</td>
</tr>
<tr>
<td>776</td>
<td>17:24:27.005</td>
<td>sched_switch</td>
<td>prev_comm=swapper/1, next_comm=netdata</td>
</tr>
<tr>
<td>777</td>
<td>17:24:27.005</td>
<td>sched_switch</td>
<td>prev_comm=netdata, next_comm=cinnamon</td>
</tr>
<tr>
<td>793</td>
<td>17:24:27.006</td>
<td>sched_switch</td>
<td>prev_comm=cinnamon, next_comm=swapper/1</td>
</tr>
<tr>
<td>794</td>
<td>17:24:27.006</td>
<td>sched_switch</td>
<td>prev_comm=swapper/1, next_comm=cinnamon</td>
</tr>
<tr>
<td>1049</td>
<td>17:24:27.025</td>
<td>sched_switch</td>
<td>prev_comm=swapper/2, next_comm=netdata</td>
</tr>
</tbody>
</table>

Process cinnamon

Process netdata
Example

Before deletion

After deletion
• **Markers**;
• **Uncertainty areas**;
• **Incoherent states**.

![Diagram](image-url)
- Markers;
- Uncertainty areas;
- Incoherent states.
View

- Markers;
- Uncertainty areas;
- Incoherent states.
Performance

Averaged on 25 executions of the analysis module
Trace777 : 2179 events, 1 lost event (164KB)
Trace1345-1360 : 40918 events, 16 lost events (2.6M)
Conclusion

New informative elements displayed on the view

Help user be aware of the uncertainty of the results

No need for special tools
Future work

Event reconstruction

Probabilistic model of events

Assess the solution on realistic use cases

Continuous work on improving the algorithms and the view → scalability
Any questions?

marie.martin@polymtl.ca

github : MMartin5