Combining Model-Based Testing and Machine Learning

Roland GROZ
LIG, Université de Grenoble, France

TAROT Summer School 2009
Acknowledgments

- Muzammil Shahbaz
- Keqin Li (now with SAP Research)
- Alexandre Petrenko (CRIM, Canada)

- Doron Peled (seminal paper BBC), David Lee, Khaled El Fakih…
Outline

- MBT and software engineering trends
- Machine Learning for test automata
- Integration: MBT and ML, integration testing
- Experiments and case studies
- Research directions
Traditional software cycle

Specification - Validation

Implementation

Vendor

Customer

Assembling fully controlled components
MBT in software development

- Model
- Design
- Implementation
- Unit tests
- Integration tests
- System tests
- Acceptance tests
Challenge: CBSE with 3rd party

Specification - Validation

Implementation

Customer

Vendor

Assembling ill controlled components
Component Based Software Engineering

- Requirement Analysis
- High Level System Design
- Components selection & Integration

- Rapid Development
- Reuse Components
- Reduce cost
- Flexibility
- Ease of integration
Issues

Understanding the System of Black Box Components is a challenge.

How do I perform system behavioral analysis?
How do I identify integration problems?
Soft. Engineering trends

- **MDE & MBT**
  - Growing trend in some industries (e.g. embedded)
  - Derive design, code and tests (MBT)
  - Models = 1st class citizens

- **CBSE**
  - Dominant growing trend
  - Absence of (formal) models
  - Pb maintaining spec <-> model
MDE & MBT in the reverse

- MDE assumption
  - Start from model, formal spec
  - Models = 1st class citizens

- Test Driven Development (XP, Agile…)
  - Tests are spec: 1st class citizens
  - Formal models ? No way!

- Proposed approach
  - Derive models from tests, & combine with MBT
  - Reconcile Test-Driven (or code-driven) dvt with Models
Technical Goals

- **Reverse Engineering**
  - Understanding the behaviors of the black box components
    - by deriving the *formal models* of the components/system
    - Can also serve documentation purposes (tests for doc)

- **System Validation**
  - Being able to derive new systematic tests
  - Analyzing the system for anomalies / compositional problems
    - by developing a *framework for integration testing* of the system of black box components
Approach

What do we achieve?

- Models (to understand the system more "formally")
- System is validated for anomalies
Partial, incremental and approximate characterization
Objections  Answers

- Model is derived from bugged components
  - Derived tests will consider bug=feature

- Incremental: stopping criterion?
Machine learning
Various types of Machine Learning

- Artificial Intelligence (& datamining)
  - Ability to infer rules, recognize patterns
  - Learning from samples
  - E.g. neural networks

- Two major techniques
  - Statistical (bayesian) inference from collection of data -> e.g. Weka tool in testing
  - *Grammatical inference of language from theoretical computer science*
Learning languages from samples

• Finding a minimum DFA (Deterministic Finite Automaton) is NP-HARD
  – Complexity of automaton identification from given data. [E. Gold 78]

• Even a DFA with no. of states polynomially larger than the no. of states of the minimum is NP-Complete
  – The minimum consistent DFA problem cannot be approximated within any polynomial. [Pitt & Warmuth 93]

• Probably Approximately Correct (PAC)
  – A theory of the learnable. [L.G. Valiant 84]
Background Work

- **Passive Learning**
  - "Learning from given positive/negative samples"
  - NP-Complete

- **Active Learning**
  - "Learning from Queries" *(Regular Inference)*
  - **Angluin's Algorithm L* [Angluin 87]**
    - Learns Deterministic Finite Automaton (DFA) in polynomial time
  - Applied in
    - Black Box Checking [Peled 99]
    - Learning and Testing Telecom Systems [Steffen 03]
    - Protocol Testing [Shu & Lee 08]
    - ...

Dana Angluin
Yale University
Concept of the Regular Inference
(Angluin's Algorithm $L^*$)

**Assumptions:**
- The input alphabet $\Sigma$ is known
- Machine can be reset

**Complexity:** $O( |\Sigma| \cdot m \cdot n^2 )$
- $|\Sigma|$: the size of the input alphabet
- $n$: the number of states in the actual machine
- $m$: the length of the longest counterexample
Our Context of Inference

- Components having I/O behaviors
- I/O are structurally complex (parameters)
- Formidable size of input sets

Enhanced State Machine Models
- Mealy Machines
- Parameterized Machines

More adequate for complex systems
DFAs may result in transition blow up

Input Alphabet $\Sigma$

The Algorithm $L^*$

Test Strategies and heuristics
Learned Models can be used to generate tests to find discrepancies
Preliminaries

- **Mealy Machine**: $M = (Q, I, O, \delta, \lambda, q_0)$
  - $Q$: set of states
  - $I$: set of input symbols
  - $O$: set of output symbols
  - $\delta$: transition function
  - $\lambda$: output function
  - $q_0$: initial state

- Input Enabled
  - $\text{dom}(\delta) = \text{dom}(\lambda) = Q \times I$

- $Q = \{q_0, q_1, q_2, q_3\}$
- $I = \{a, b\}$
- $O = \{x, y\}$
Mealy Machine Inference Algorithm

The Algorithm $L_M^*$

**Assumptions:**
- The input set $I$ is known
- Machine can be reset
- For each input, the corresponding output is observable
Basic principles of Angluin’s algorithm (mod.)

I={a, b}

Observation Table

\[ \begin{array}{c|cc}
  & a & b \\
  \varepsilon & x & x \\
  a & y & x \\
  b & x & x \\
  aa & y & x \\
  ab & x & x \\
\end{array} \]

S (span seq) for States

S \cdot I transitions

Conjecture

\[ tr_1: a/x \]
\[ tr_2: b/x \]
\[ tr_4: b/x \]
\[ tr_3: a/y \]

Observation Table

Black Box Mealy Machine Component

• \( \varepsilon \) is an empty string
Mealy Machine Inference Algorithm $L_M^*$ (1/5)

**Concepts:**
- Closed
- Consistency

- $\varepsilon$ is an empty string

**Observation Table ($S_M \cdot E_M \cdot T_M$)**

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$a$</td>
<td>$y$</td>
<td>$x$</td>
</tr>
<tr>
<td>$b$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

**Black Box Mealy Machine Component**

**Output Queries:**
- $s \cdot e$, $s \in (S_M \cup S_M \cdot I)$, $e \in E_M$
- $= a / x$

$I=\{a, b\}$
Mealy Machine Inference Algorithm $L_M^*$ (2/5)

Concept: Closed

<table>
<thead>
<tr>
<th>$S_M$ • $I$</th>
<th>$E_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>$a$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>$x$</td>
</tr>
<tr>
<td>$a$</td>
<td>$y$</td>
</tr>
<tr>
<td>$b$</td>
<td>$x$</td>
</tr>
<tr>
<td>aa</td>
<td>$y$</td>
</tr>
<tr>
<td>ab</td>
<td>$x$</td>
</tr>
</tbody>
</table>

Concepts:
- Closed: All the rows in $S_M$ • $I$ must be equivalent to the rows in $S_M$
  - Same behaviour = known state
- Consistency

$\epsilon$ is an empty string
Mealy Machine Inference Algorithm $L_M^*$ (3/5)

Making Conjecture

<table>
<thead>
<tr>
<th>$S_M$</th>
<th>$I$</th>
<th>$E_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>a</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>aa</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Counterexample:

- a b a b b a a
- component's response: x x x x x x y
- conjecture's response: x x x x x x x

Black Box Mealy Machine Component

$tr_1$: a/x

$tr_2$: b/x

$tr_3$: a/y

$tr_4$: b/x
Mealy Machine Inference Algorithm $L_M^*$ (4/5)

Processing Counterexamples

<table>
<thead>
<tr>
<th>$S_M$</th>
<th>$E_M$</th>
<th>$S_{M\cdot I}$</th>
<th>$E_{M'}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>a</td>
<td>y</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>aa</td>
<td>y</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ab</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**Observation Table ($S_M, E_M, T_M$)**

**Counterexample:** a b a b b a a

**Method:**

Add all the prefixes of the counterexample to $S_M$
Mealy Machine Inference Algorithm $L_M^*$ (5/5)

Concept: Consistency

<table>
<thead>
<tr>
<th>Concept</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>a</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>ab</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>aba</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>abab</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ababb</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ababba</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ababbaa</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>aa</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>abb</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>abaa</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ababa</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>ababbb</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Observation Table ($S_M, E_M, T_M$)

Concepts:
- Closed
- Consistency: All the successor rows of the equivalent rows must also be equivalent

Consistency check can be avoided if all rows in $S_M$ are inequivalent

The rows in $S_M$ become equivalent due to the method of processing counterexamples in the table.
New Method for Processing Counterexamples
(The Algorithm $L_M^+$)

**Observation Table ($S_M, E_M, T_M$)
before processing counterexample**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>a</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>aa</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>ab</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**Counterexample**

a b a b b a a

Add all the suffixes to $E_M$

**Observation Table ($S_M, E_M, T_M$)
after processing counterexample**

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>aa</th>
<th>baa</th>
<th>bbaa</th>
<th>abbaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>x</td>
<td>x</td>
<td>xy</td>
<td>xxx</td>
<td>xxxy</td>
<td>xyyyy</td>
</tr>
<tr>
<td>a</td>
<td>y</td>
<td>x</td>
<td>yy</td>
<td>xxx</td>
<td>xxxx</td>
<td>yyyy</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xyy</td>
<td>xxxx</td>
<td>xxxyy</td>
</tr>
<tr>
<td>aa</td>
<td>y</td>
<td>x</td>
<td>xx</td>
<td>xxx</td>
<td>xxxy</td>
<td>xxxxx</td>
</tr>
<tr>
<td>ab</td>
<td>x</td>
<td>x</td>
<td>yy</td>
<td>xxx</td>
<td>xxxx</td>
<td>yyyy</td>
</tr>
</tbody>
</table>

All rows remain inequivalent (inconsistency never occurs)
Comparison of the two Methods

Total Output Queries in $L^+_M$: 64

<table>
<thead>
<tr>
<th></th>
<th>b</th>
<th>aa</th>
<th>baa</th>
<th>bbab</th>
<th>abbaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>x</td>
<td>x</td>
<td>xy</td>
<td>xxx</td>
<td>xxxx</td>
</tr>
<tr>
<td>a</td>
<td>y</td>
<td>x</td>
<td>yy</td>
<td>xxx</td>
<td>xxxx</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>ab</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxx</td>
<td>xxxx</td>
</tr>
<tr>
<td>aa</td>
<td>y</td>
<td>x</td>
<td>yy</td>
<td>xxx</td>
<td>xxxy</td>
</tr>
<tr>
<td>ba</td>
<td>x</td>
<td>x</td>
<td>xy</td>
<td>xxx</td>
<td>xxxy</td>
</tr>
<tr>
<td>bb</td>
<td>x</td>
<td>x</td>
<td>xy</td>
<td>xxx</td>
<td>xxxy</td>
</tr>
<tr>
<td>aba</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>abb</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
</tbody>
</table>

Final Observation Table ($S_M^pE_M^pT_M$) after processing counterexample according to $L^+_M$

Total Output Queries in $L^*_M$: 86

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>x</th>
<th>yy</th>
<th>xxx</th>
<th>xxxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>aba</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxx</td>
</tr>
<tr>
<td>abab</td>
<td>y</td>
<td>x</td>
<td>yy</td>
<td>xxx</td>
<td>xxxx</td>
</tr>
<tr>
<td>ababba</td>
<td>x</td>
<td>x</td>
<td>xy</td>
<td>xxx</td>
<td>xxxy</td>
</tr>
<tr>
<td>ababbaa</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>b</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>aa</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>abb</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>abaa</td>
<td>y</td>
<td>x</td>
<td>yy</td>
<td>xxx</td>
<td>xxxx</td>
</tr>
<tr>
<td>ababa</td>
<td>x</td>
<td>x</td>
<td>xy</td>
<td>xxx</td>
<td>xxxy</td>
</tr>
<tr>
<td>ababbb</td>
<td>x</td>
<td>x</td>
<td>xy</td>
<td>xxx</td>
<td>xxxy</td>
</tr>
<tr>
<td>ababbab</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>ababbaaa</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
<tr>
<td>ababbaab</td>
<td>x</td>
<td>x</td>
<td>xx</td>
<td>xxy</td>
<td>xxxy</td>
</tr>
</tbody>
</table>

Final Observation Table ($S_M^pE_M^pT_M$) after processing counterexample according to $L^*_M$
Comparison of the two Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_M^*$</td>
<td>$O(</td>
</tr>
<tr>
<td>$L_M^+$</td>
<td>$O(</td>
</tr>
</tbody>
</table>

- $I$: the size of the input set
- $n$: the number of states in the actual machine
- $m$: the length of the longest counterexample
# Experiments with Edinburgh Concurrency Workbench

<table>
<thead>
<tr>
<th>Examples</th>
<th>Size of Input set</th>
<th>No. of States</th>
<th>No. of Output Queries for $L_M^*$</th>
<th>No. of Output Queries for $L_M^+$</th>
<th>Reduction Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>3</td>
<td>6</td>
<td>48</td>
<td>48</td>
<td>0</td>
</tr>
<tr>
<td>ABP-Lossy</td>
<td>3</td>
<td>11</td>
<td>764</td>
<td>340</td>
<td>1,22</td>
</tr>
<tr>
<td>Peterson2</td>
<td>3</td>
<td>11</td>
<td>910</td>
<td>374</td>
<td>1,43</td>
</tr>
<tr>
<td>Small</td>
<td>6</td>
<td>11</td>
<td>462</td>
<td>392</td>
<td>0,18</td>
</tr>
<tr>
<td>VM</td>
<td>6</td>
<td>11</td>
<td>836</td>
<td>392</td>
<td>1,13</td>
</tr>
<tr>
<td>Buff3</td>
<td>3</td>
<td>12</td>
<td>680</td>
<td>369</td>
<td>1,24</td>
</tr>
<tr>
<td>Shed2</td>
<td>6</td>
<td>13</td>
<td>824</td>
<td>790</td>
<td>0,04</td>
</tr>
<tr>
<td>ABP-Safe</td>
<td>3</td>
<td>19</td>
<td>2336</td>
<td>764</td>
<td>2,1</td>
</tr>
<tr>
<td>TMR1</td>
<td>6</td>
<td>19</td>
<td>1396</td>
<td>1728</td>
<td>-0,19</td>
</tr>
<tr>
<td>VMnew</td>
<td>4</td>
<td>29</td>
<td>3696</td>
<td>1404</td>
<td>0,86</td>
</tr>
<tr>
<td>CSPROT</td>
<td>6</td>
<td>44</td>
<td>4864</td>
<td>3094</td>
<td>0,67</td>
</tr>
</tbody>
</table>

### Graph

![Graph showing the relationship between Examples and $L_M^*$ and $L_M^+$](image)
System Architecture

- System of communicating Mealy Machine Components
- Components are deterministic and input-enabled
- System has *External* and *Internal* i/o interfaces
  - External interface is controllable
  - External and Internal interfaces are observable
- Single Message in Transit and Slow Environment
Learning & Testing Framework

**Step 1:** Learn Models

**Step 2(a):** Compose Models

**Step 2(b):** Analyze Product

**Step 2(c):** Confirm Problem on System

**Step 3:** Refine Models

**Step 4:** Generate Tests from Product

**Step 5:** Resolve Discrepancy (exception, crash, out of memory,...?)

- [compositional problem]
- [discrepancy as counterexample]
- [problem confirmed]
- [error found]
- [problem as counterexample]
- [no discrepancy]
- [termination]

- No Compositional Problems
Contributions

1. Mealy Machine Inference
   - Improvements in the basic adaptation from the Angluin's algorithm

2. Parameterized Machine Inference

3. Framework of *Learning and Testing* of integrated systems of black box Mealy Machine components

4. Tool & Case Studies (provided by Orange Labs)
The tool **RALT**

(Rich Automata Learning and Testing)
Case Studies

1. Concurrency Workbench

2. Air Gourmet

3. Nokia 6131, N93, Sony Ericsson W300i
   - Experimented with Media Player

1. Domotics (Orange Labs' Smart Home Project)
   - Devices: Dimmable Light, TV, Multimedia Systems
# Air Gourmet

Goal: Learning & Testing the System

<table>
<thead>
<tr>
<th>Components</th>
<th>Size of Input Set</th>
<th>No. of States</th>
<th>No. of Errors</th>
<th>Error Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passenger Check-in</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Flight Reservation</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>NPE, IIE, Date Parsing Exception</td>
</tr>
<tr>
<td>Meals Manager</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>NPE, IIE, IAE</td>
</tr>
</tbody>
</table>

**NPE**: Null Pointer Exception, **IIE**: Invalid Input Exception, **IAE**: Illegal Input Exception
Nokia 6131
Goal: Learning the behaviors of the Media Player

{ Play, Pause, Stop }

RALT

Music P1.wav
Music P2.wav
Nokia 6131 / N93 / Sony Ericsson W300i

Goal: Learning the behaviors of the Media Player

- p1.start(p1.wav) / p1.STARTED
- p1.stop() / p1.stopped
- p1.close() / p1.CLOSED
- p1.stop() / p1.STOPPED

- p2.start(p2.wav) / Exp
- p2.close / p2.CLOSED
- p2.start(p2.wav) / p2.STARTED
- p2.stop() / p2.STOPPED

Nokia 6131 (PFSM)

Nokia 93 / Sony Ericsson W300i (PFSM)
Domotics
Goal: Learning the interactions of the devices

<table>
<thead>
<tr>
<th>Device</th>
<th>Size of Input Set</th>
<th>No. of States</th>
<th>Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light (ProSyst)</td>
<td>4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Media Renderer (Philips)</td>
<td>5</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>Domotics (Interaction Model)</td>
<td>9</td>
<td>16</td>
<td>30</td>
</tr>
</tbody>
</table>
Conclusion

- Reverse Engineering Enhanced State Models
  - Improved Mealy machine inference
  - Parameterized Machine Inference

- Framework for Learning & Testing of Integrated Systems of Black Box Mealy machines

- The tool *RALT* that implements the reverse engineering and integration testing framework

- Experiments with real systems in the context of Orange Labs.
Perspectives

- **Work in Progress**
  - Approach for detecting sporadic errors
  - Learning Nondeterministic Automata

- **Future Work**
  - Test Generation for Model Refinements
  - Framework for PFSM models
Behind the Curtain
DoCoMo: A Motivational Example in Orange Labs

Hidden Behavior:
User's scores are uploaded to the server through web

Hidden Interaction:
The Game component interacts with the Web component for connection
Mealy Machine Quotients

Let $\Phi$ be a set of strings from $I$ then

- the states $s1$ and $s2$ are $\Phi$-equivalent if they produce same outputs for all the strings in $\Phi$
- A quotient based upon $\Phi$-equivalence is called $\Phi$-quotient

$\Phi = \{a, b, ab, ba, bb, bba\}$

$q_0$ and $q_2$ are $\Phi$-Equivalent
$q_1$ and $q_3$ are $\Phi$-Equivalent
Relation between the Conjecture and the Black Box Machine

Closed (and Consistent) Observation Table ($S_M, E_M, T_M$)

Conjecture from the Observation Table ($S_M, E_M, T_M$)

Black Box Mealy Machine

<table>
<thead>
<tr>
<th>$S_M \cdot I$</th>
<th>$E_M$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$a$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$aa$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
<tr>
<td>$ab$</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
</tbody>
</table>

$E_M$-Quotient
Objections    Answers
Contributions

1. Mealy Machine Inference
   - Improvements in the basic adaptation from the Angluin's algorithm

1. Parameterized Machine Inference

1. Framework of *Learning and Testing* of integrated systems of black box Mealy Machine components

2. Tool & Case Studies (provided by Orange Labs)
Parameterized Finite State Machine (PFSM)
PFSM Algorithm $L_P^*$ (a view)

$\delta$ (transition function): $\delta(q, a) = q'$

$\delta$ is defined for each state $q$ and input symbol $a$.

$\delta(q_0, a) = q_1$ for $a \in \{a, b\}$

$\delta(q_1, b) = q_2$ for $b \in \{x \leq 5\}$

$\delta(q_2, b) = q_0$ for $b \in \{x > 5\}$

$\delta(q_3, b) = q_0$ for $b \in \{x \leq 1\}$

$\delta(q_4, a) = q_1$ for $a \in \{a, b\}$

$\delta(q_4, b) = q_2$ for $b \in \{x \leq 1\}$

$\delta(q_5, a) = q_1$ for $a \in \{a, b\}$

$\delta(q_5, b) = q_2$ for $b \in \{x \leq 1\}$

Observation Table ($S_{p}, R, E_{p}, T_{p}$)

$$
I = \{a, b\}, D_{1} = \mathbb{N} \cup \{\bot\}
$$

Black Box PFSM Component
Contributions

1. Mealy Machine Inference
   - Improvements in the basic adaptation from the Angluin's algorithm

2. Parameterized Machine Inference

1. Framework of *Learning and Testing* of integrated systems of black box Mealy Machine components

2. Tool & Case Studies (provided by Orange Labs)