A conformance relation for model-based testing of PLC

Anaïs Guignard
LURPA, ENS Cachan
Summary

I / Introduction

II / Test sequence construction and execution

III / A conformance relation for MIC test sequences

IV / Conclusions and perspectives
Summary

I / Introduction
  • Reminder on conformance testing
  • Claim
  • Assumptions
  • Notations
  • Treated example

II / Test sequence construction and execution

III / A conformance relation for MIC test sequences

IV / Conclusions and perspectives
Reminder – Validation

Validation of an implementation [Boehm,79]:

- Are we doing the right product?

Focus on conformance test
Reminder – Conformance test

Conformance test:

Based on:

- Knowledge of the formalized specification
- Generation of a test sequence depending on a given test objective
- Verdict for the whole test sequence depending on the conformance relation
- No knowledge about the plant
Relations to verify

Test objective, usually given as coverage criterion:

- Path coverage, … (Software)
- Output, logic gates coverage, … (logic circuits)
- States, Input combinations, transitions, … (functional test)

Conformance relation (also called implementation relation):

- A formal relation between implementation and specification model
- Expresses the correctness of the implementation with respect to specifications
Conformance relation

Example of conformance relations:

- For labeled transition system: [Tretmans,96]
  - $\text{ioco } S = \forall \sigma \in \text{Straces}(S): \text{out}(I \text{ after } \sigma) \subseteq \text{out}(S \text{ after } \sigma)$

- For timed automata: [Styp,10], [Nunez,02]
  - Timed LTS: $\text{ioco } S = \forall \sigma \in \text{Straces}(S): \text{out}_t(I \text{ after}_t \sigma) \subseteq \text{out}_t(S \text{ after}_t \sigma)$
  - Timed FSM: $\text{conf}_a S \text{ iff conf}_n t S \text{ and } \forall e \in NTEvol(I) \cap NTEvol(S), \forall t, (e, t) \in TEvol(I) \Rightarrow (e, t) \in TEvol(S)$

- For Mealy machine:
  - $\text{conf } S \text{ iff } \forall e \in \sigma_{test}, \text{ the observed } (I_{obs}, O_{obs}) \text{ is such as } \delta(s_w, I_{obs}) = s_d \text{ and } \lambda(s_w, I_{obs}) = O_{obs}$
Claim

• Usual conformance relations are based on models of the implementation

• Conformance test is performed on a Programmable Logic Controllers (PLC)

• We claim to propose a conformance relation that takes in count the features of the PLC as the cyclic reading of the inputs values
Assumptions

On the specification model:
• Non timed
• Mealy machine
• Complete and deterministic
• Without transient evolution (two following transitions that are not self-loop with the same input condition)
• States distinguishable by the output emission

On the observation:
• All Input/Output changes are detected
Notations

$V_I$ Set of input variables
$V_O$ Set of output variables

A Mealy machine is a 6 - tuple $(I_M, O_M, S, s_{init}, \delta, \lambda)$ where:

• $I_M$ is the input alphabet
• $O_M$ is the output alphabet
• $S$ is the set of states
• $s_{init}$ is the initial state
• $\delta: S \times I_M \rightarrow S$ is the transition function
• $\lambda: S \times I_M \rightarrow O_M$ is the output function

$\sigma = (..., (I_{obs}, O_{obs}), ...)$ Observation sequence with $I_{obs} \in I_M$ and $O_{obs} \in O_M$
Example of specification model

2 inputs: $V_I = \{a, b\}$

1 output: $V_O = \{o\}$
Summary

I / Introduction

II / Test sequence construction and execution
  • MIC test sequence
  • Desynchronization phenomenon and consequences
  • SIC test sequence
  • SIC/MIC test sequence

III / A conformance relation for MIC test sequences

IV / Conclusions and perspectives
Test step

Transition to test:

Elementary test step: $et = (s_w, I^j, s_d, O^j)$
- Remark: One test step test both the transition and the self-loop

Test sequence: $TS = (et_1, ..., et_n)$
Test sequence construction and execution

Test sequence

Test sequence with:

- Minimal-length test sequence

\[ TS_{MIC} = ((s_1, \overline{a} \cdot \overline{b}, s_1, \overline{o}), (s_1, a \cdot b, s_2, o), (s_2, \overline{a} \cdot \overline{b}, s_2, o), (s_2, a \cdot \overline{b}, s_1, \overline{o}), (s_1, a \cdot b, s_2, o), (s_2, \overline{a} \cdot b, s_1, \overline{o})) \]

There are some Multiple – Input – Change (MIC) test steps.

- For example: \((s_1, \overline{a} \cdot \overline{b}, s_1, \overline{o}), (s_1, a \cdot b, s_2, o)\) need \(a\) and \(b\) to synchronously change to True

This may be wrongly treated by the PLC.
Desychronization phenomenon

Due to the Cyclic I/O scanning of the PLC

Synchronous changes of inputs can be read on different PLC cycles
Frequency on test execution

Experiment [Provost,10]:
- 8 simultaneous input changes from False to True then True to False
- Inputs allocated on two I/O modules
- 20 000 drawings

<table>
<thead>
<tr>
<th>Input changes \ PLC period</th>
<th>10 ms</th>
<th>20 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>On main module</td>
<td></td>
<td></td>
</tr>
<tr>
<td>True to False</td>
<td>0,022 %</td>
<td>0,014 %</td>
</tr>
<tr>
<td>False to True</td>
<td>0,886 %</td>
<td>0,455 %</td>
</tr>
<tr>
<td>On secondary module</td>
<td></td>
<td></td>
</tr>
<tr>
<td>True to False</td>
<td>0,040 %</td>
<td>0,020 %</td>
</tr>
<tr>
<td>False to True</td>
<td>0,993 %</td>
<td>0,550 %</td>
</tr>
<tr>
<td>On both modules</td>
<td></td>
<td></td>
</tr>
<tr>
<td>True to False</td>
<td>39,62 %</td>
<td>20,23 %</td>
</tr>
<tr>
<td>False to True</td>
<td>43,46 %</td>
<td>21,89 %</td>
</tr>
</tbody>
</table>
SIC test sequence

Test sequence [Provost, 10]:
- Only Single – Input – Change test steps
- Minimal length

\[ TS_{SIC} = ( (s_1, \overline{a} \cdot \overline{b}, s_1, \overline{o}), (s_1, a \cdot \overline{b}, s_1, \overline{o}), (s_1, a \cdot b, s_2, o), (s_2, a \cdot \overline{b}, s_1, \overline{o}), (s_1, a \cdot b, s_2, o), (s_2, \overline{a} \cdot b, s_1, \overline{o}) ) \]

Test step \((s_2, \overline{a} \cdot \overline{b}, s_2, o)\) is missing

Conclusion:
- Not always possible to guarantee the test objective with a SIC sequence
Minimal MIC test sequence

Solution SIC – MIC [Provost,10]:

- Minimal-length SIC sequence
- Followed by non-SIC-testable test steps

\[
T_{S_{mixed}} = ((s_1, \overline{a}. \overline{b}, s_1, \overline{o}), (s_1, a. \overline{b}, s_1, \overline{o}), (s_1, a. b, s_2, o),
(s_2, a. \overline{b}, s_1, \overline{o}), (s_1, a. b, s_2, o), (s_2, \overline{a}. b, s_1, \overline{o}),
(s_1, a. b, s_2, o), (s_2, \overline{a}. \overline{b}, s_2, o))
\]

Discussion:

- Reduces the amount of MIC test steps some can not be avoided
- The conformance relation must take them in count
Summary

I / Introduction

II / Test sequence construction and execution

III / A conformance relation for MIC test sequences

• Consequences of desynchronization on test
• Definition of a conformance relation
• Illustration
• Pursuit of test execution

IV / Conclusions and perspectives
Expected behavior

Initial situation:
- State $s_2$ active
- Input combination $a.b$

Test step:
- $et = (s_2, \overline{a}.\overline{b}, s_2, o)$

Expected sequence:
- $\sigma = ((a.b, o), (\overline{a}.\overline{b}, o), (\overline{a}.\overline{b}, o), ...)$
Effective behavior

I/O observed outside the PLC

Expected sequence:
• \( \sigma = \{(a \cdot b, o), (\bar{a} \cdot \bar{b}, o), (\bar{a} \cdot \bar{b}, o), \ldots \} \)

PLC read/updated sequence:
• \( \sigma = \{(a \cdot b, o), (a \cdot \bar{b}, \bar{o}), (\bar{a} \cdot \bar{b}, \bar{o})\} \)

Observed sequence:
• \( \sigma = \{(a \cdot b, o), (\bar{a} \cdot \bar{b}, \bar{o}), (\bar{a} \cdot \bar{b}, \bar{o})\} \)
A conformance relation for MIC test sequences

Idea

The observed sequence is not the expected one

However, this implementation conforms to the specification

It should not be rejected by the conformance relation.
Conformance relation

Let \( e_t(c) = (s_b; I^j; s; O^j) \) be the current test step, Let \( e_t(p) = (s_a; I^i; s_b; O^i) \) be the previous test step.

The implementation conforms to the specification if for every test step there is:

Either:

- If \( s_b \neq s_c \): \( \exists k \in \mathbb{N}^* \text{ such as } k < n \text{ and:} \)
  - If \( k > 1 \): \( \forall l \in \mathbb{N}^* \text{ such as } l < k, O_{obs_l} = O^i \)
  - \( O_{obs_k} = O^j \)
  - \( \forall m \in \mathbb{N}^* \text{ such as } k < m \leq n, O_{obs_m} = O^j \)
- If \( s_b = s_c \): \( \forall k \in \mathbb{N}^* \text{ such as } k \leq n, O_{obs_k} = O^j \)

Or

- \( \exists k < n - 1 \) such as:
  - If \( k > 1 \): \( \forall l \in \mathbb{N}^* \text{ such as } l < k, O_{obs_l} = O^i \)
  - It exists \( l^x \in I^x_M \) such as:
    - \( (l^x \setminus I^i \cup I^i \setminus l^x) \subset (l^i \setminus I^i \cup I^i \setminus l^i) \)
    - \( \lambda(s_b, l^x) = O_{obs_k} \)
  - Let \( s = \delta(s_b, l^x) \) be the downstream state of the transition,
  - It exists a transition such as \( \lambda(s, I^l) = O_{obs_{k+1}} \)
  - And \( \forall m \in \mathbb{N}^* \text{ such as } k + 1 < m \leq n, O_{obs_m} = O_{obs_{k+1}} \)

Test of the firing of a transition

Test of a self-loop

Test on which input changes are seen asynchronously:
- Checks if a transition exists from the current state associated to the observed output
- Checks if the input condition is a subset of input changes
- Checks if the next PLC cycle correctly takes in count all the input changes
A conformance relation for MIC test sequences

Example of a correct implementation

Initial situation:
- State $s_2$ active
- Input combination $a.b$

Test step:
- $et = (s_2, \bar{a}.\bar{b}, s_2, o)$

Expected sequence:
- $\sigma = (a.b, o), (\bar{a}.\bar{b}, o), (\bar{a}.\bar{b}, o), \ldots$

Observed sequence:
- $\sigma = (a.b, o), (\bar{a}.\bar{b}, o), (\bar{a}.\bar{b}, o)$
Example of an incorrect implementation

Initial situation:
- State $s_2$ active
- Input combination $a \cdot b$

Test step:
- $et = (s_2, \bar{a} \cdot \bar{b}, s_2, o)$

Expected sequence:
- $\sigma = ((a \cdot b, o), (\bar{a} \cdot \bar{b}, o), (\bar{a} \cdot \bar{b}, o), ...)$

Observed sequence:
- $\sigma = ((a \cdot b, o), (\bar{a} \cdot \bar{b}, o), (\bar{a} \cdot \bar{b}, o))$
Pursue of the test execution

After a desynchronization phenomenon:

• There is no guarantee that the active state is the expected state
• The active state will not correspond to the one in the next test step
• It is necessary to modify the test sequence to continue the test

Possible solutions:

• Recomputation of the whole sequence
• Finding a previous test step from which to restart the test
• Finding a SIC path to come back to the treated test step
Summary

I / Introduction

II / Test sequence construction and execution

III / A conformance relation for MIC test sequences

IV / Conclusions and perspectives
Conclusions and perspectives

Conclusion:
- To meet the test objective, it is not always possible to prevent MIC test steps
- Asynchronous read of synchronous input changes cannot be ignored in the conformance relation
- A new conformance relation has been defined
- To be published in WODES’14:

Perspectives:
- Choice and implementation of a method to pursue test execution
- Adaptation to closed-loop validation methods (to be submitted at ETFA’14)
Bibliography

[Boehm,79]:

[Tretmans,96]:
J. Tretmans: *Test Generation with Inputs, Outputs and Repetitive Quiescence*, Software concept and tool 17, 1996

[Styp,10]:

[Nunez,02]:
M. Nunez, I. Rodriguez, *Encoding PAMR into (Timed) EFSMs*, Formal Techniques for Networked and Distributed Sytems — FORTE, 2002

[Provost,10]: