Model-Based Testing Real-Time and Interactive Music Systems

Thesis defended on 11/10/16

Poncelet Sanchez Clement,
Florent Jacquemard

SYNCHRON 2016

Team: RepMus
Score-Based Interactive Music Systems

discrete inputs/outputs

evt

e → a

input

output

IMS

Mixed Score

hard time

synchronous

act
Anthèmes 2
pour violon et dispositif électronique (1997)

Libre

\( j = 92 \)

ban. (archet normal)

\( j = 66 \)

Pierre Boulez
(*1925)

Violon

Spatialization

F -11/-18/-18/2.0

Inf. Rev.

reverb. time: 60''

Spatialization

F -11/-18/-18/2.0

= 90 msec.

Sampl. IR

reverb. time: 60''

Spatialization

F -11/-18/-18/2.0

= 93 msec.

Sampler

Spatialization

MR -4/-12/24/2.0

Mixed Score
Anthèmes 2
pour violon et dispositif électronique (1997)

Pierre Boulez
(*1925)

Mixed Score

Interpretation
Anthèmes 2
pour violon et dispositif électronique (1997)

Pierre Boulez
(*1925)

Mixed Score
Anthèmes 2
pour violon et dispositif électronique (1997)

Pierre Boulez
(*1925)

Mixed Score

Interpretation

infinites possibilities de performances
Timed Conformance Testing

Set of relevant inputs → Set of corresponding implementation outputs → Computation of expected outputs

Timed conformance
Manual Testing IMS

Manual Methods
- Rehearsals

- Test for one performance
- Time costly
- Not precise

environment

evt

act

IMS
Model-Based Testing IMS

- ‘Exhaustive’ generation
- Fast forward execution (virtual clocks)
- Precise:
  - Automated comparison
  - Formal conformance criteria
  - Informative feedbacks
Model-Based Testing IMS

Environment

Implementation Under Test

\[
\begin{align*}
\text{Environment} & \quad \text{evt} \\
\text{Implementation Under Test} & \\
\end{align*}
\]

Bound performances

Automatic construction

\[\mathcal{E} \xrightarrow{\text{act}} \mathcal{S} \]
Outline

1. Objectives

2. Testing Framework

3. Interactive Real-Time Model
Testing Framework Overview

Mixed Score

Build
- from high level to model

Construction:

Model- Based Testing:
- from model to verdict

Online Approach

Offline Approach

Verdict

Mixed Score

Model-Based Testing:
- from model to verdict

Testing Framework Overview
Testing Approaches

1. Inputs Generation
   - Compute Expected Outputs

2. Model + Mixed Score
   - Testing Approaches

3. (Simulation) Compute Real Outputs
   - Timed conformance

4. (Execution) Compute Real Outputs
   - Outputs Generation

Model + Mixed Score
Outline

1. Objectives

2. Testing Framework

3. Interactive Real-Time Model
Input / Output formalisation
A timed trace is a tuple \( \langle s, t, p \rangle \):

- \( s \): symbol
- \( t \): timestamp in time unit
- \( p \): pace in time unit per minute

**Definition:**

\[ e_1, 0, 120 \]
**Timed Traces**

A timed trace is a tuple \(<s, t, p>\):

- **s**: symbol
- **t**: timestamp in time unit
- **p**: pace in time unit per minute

**Definition:**

Specified inputs

\(<e_1, 0, 120>, e_2, 2, 120>
Timed Traces

Definition:

A timed trace is a tuple \(<s, t, p>:\n
- \(s\): symbol
- \(t\): timestamp in time unit
- \(p\): pace in time unit per minute

tempo: 120bpm

Specified inputs

\(<e_1, 0, 120>, <e_2, 2, 120>, <e_3, 2.33, 120>\)
Timed Traces

Definition:
A timed trace is a tuple \(<s, t, p>\):
- \(s\): symbol
- \(t\): timestamp in time unit
- \(p\): pace in time unit per minute

Specified inputs:
\(<e_1, 0, 120><e_2, 2, 120><e_3, 2.33, 120><e_4, 2.66, 120><e_5, 3, 120><e_6, 6, 120>\)

\(<e_1, 0, 120><e_2, 2, 120><e_4, 2.66, 120><e_5, 3, 120><e_6, 6, 120>\)

\(<e_1, 0, 120><e_2, 1.9, 120><e_4, 2.76, 120><e_5, 3.2, 120><e_6, 5.9, 120>\)

\(<e_1, 0, 119><e_2, 1.9, 80.9><e_4, 2.76, 114><e_5, 3.2, 115.3><e_6, 5.9, 119>\)

Errors:
\(<e_1, 0, 120><e_2, 1.9><e_4, 2.66, 120><e_5, 3, 120><e_6, 6, 120>\)

Variations:
\(<e_1, 0, 120><e_2, 1.9><e_4, 2.76, 120><e_5, 3.2, 120><e_6, 5.9, 120>\)
Timed Traces

Specified inputs

< e, 0, 120 \times e_2, 2, 120 \times e_3, 2.33, 120 \times e_4, 2.66, 120 \times e_5, 3, 120 \times e_6, 6, 120 >

< e, 0, 120 \times e_2, 2, 120 > ! < e_4, 2.66, 120 \times e_5, 3, 120 \times e_6, 6, 120 >

< e, 0, 120 \times e_2, 1.9, 120 > < e_4, 2.76, 120 \times e_5, 3.2, 120 \times e_6, 5.9, 120 >

< e, 0, 119 \times e_2, 1.9, 80.9 > < e_4, 2.76, 114 \times e_5, 3.2, 115.3 \times e_6, 5.9, 119 >
Timed Conformance Testing

Set of relevant inputs

Set of corresponding implementation outputs

Computation of expected outputs

Timed conformance
Models

Environment Model

\[ \mathcal{E} \]

Bound performances

System Model

\[ S \]

Model = \( E + S \)

Compute expected output

Interactive Real-Time Model

Timed Automata with Input-Output

TA aspects

Synchronous aspects

model-checking / decidability
Labelled Transition System

System Specification

Input/Output

e  a

e_1?  a_1!  a_2!

Jan Tretmans.
Model Based Testing with Labelled Transition Systems.
Formal Methods and Testing, an outcome of the FORTEST.

Model Based Testing.
Software and Systems Safety - Specification and Verification.
Labelled Transition System

Simulation

\[ e_1 ? \rightarrow a_1 ! \rightarrow a_2 ! \]

Jan Tretmans.
Model Based Testing with Labelled Transition Systems.
Formal Methods and Testing, an outcome of the FORTEST.

Model Based Testing.
Software and Systems Safety - Specification and Verification.
Labelled Transition System

Simulation

Jan Tretmans. Model Based Testing with Labelled Transition Systems. Formal Methods and Testing, an outcome of the FORTEST.

Labelled Transition System

Simulation

Jan Tretmans.
Model Based Testing with Labelled Transition Systems.
Formal Methods and Testing, an outcome of the FORTEST.

Model Based Testing.
Software and Systems Safety - Specification and Verification.
Labelled Transition System

Simulation

Jan Tretmans. 
Model Based Testing with Labelled Transition Systems. 
Formal Methods and Testing, an outcome of the FORTEST.

Model Based Testing. 
Software and Systems Safety - Specification and Verification.
Timed Automata with Input-Output

System Specification

Clock value: \( C_i = \mathbb{R}_+ \).

Finite set of clocks valued on \( \mathbb{R}_+ \).

Abstract Time

Same rate

Urgent locations

Restricted with guards and invariants

Reset with affectations

---

R. Alur and D. Dill.

A. David, K.G. Larsen, S. Li, M. Mikucionis, B. Nielsen.
Testing real-time systems under uncertainty. FMCO’10.
Interactive Real-Time Model (IRTM)

System Specification

- \( e_1 ? \rightarrow a_1 ! \rightarrow 0.125 \text{ m.u} \rightarrow a_2 ! \)
- \( e_2 ? \rightarrow a_3 ! \rightarrow 93 \text{ ms} \rightarrow a_4 ! \rightarrow 93 \text{ ms} \rightarrow a_5 ! \)

- Extended aspects
  - Multiple time units
  - Alternation

- TA aspects
  - Clock constraints
    - Transition Discrete/Temporal

\( m.u: \text{musical time unit} \)
\( m.s: \text{mini seconds} \)
IRTM: System Model

State: \( <t, n> \) \[ \text{controls} \] \{ symbols \}

control: \( C_i = \mathbb{R}_+ \)

Dense time


IRTM: System Model

Simulation

State: \( \langle 0, 0 \rangle \) \[ \begin{bmatrix} C1 & 0 \end{bmatrix} \} \}
IRTM: System Model

State: $\langle 0, 1 \rangle \left[ C_1 = 0 :: C_2 = 0 \right] \{ \}$
Cooperative Scheduling

Simulation

State: \( \langle 0, 1 \rangle \left[ \begin{array}{c} c_1=0 \\ c_2=0 \end{array} \right] \) \{ \}
IRTM: System Model

Simulation

State: $\langle 0, 1 \rangle \left[ C_1=0 :: C_2=0 \right] \{ \}$

Synchronous aspect

Transition
Discrete/Temporal

End of logical instant
IRTM: System Model

State: $\langle 0, 1 \rangle \left[ \begin{array}{c} C_1=0 ; C_2=0 \end{array} \right] \{ e_1 \}$
IRTM: System Model

State: \( \langle 0, 2 \rangle \left[ \begin{array}{c} C_1 = 0 \; ; \; C_2 = 0 \end{array} \right] \{ e_1 \} \)
IRTM: System Model

Simulation

State: $\langle 0, 2 \rangle \left[ C_1=0 ; C_2=0 \right] \{ e_1 \}$
IRTM: System Model

Simulation

Extended aspects

Priorities

State: $\langle 0, 2 \rangle \left[ \begin{array}{c} c_1=0 \\ c_2=0 \end{array} \right] \{ e_1 \}$
IRTM: System Model

State:  \( \langle 0, 3 \rangle \ \left[ C_1=0 \ ; \ C_2=0 \right] \ \{ e_1 \} \)
IRTM: System Model

Simulation

State: \(<0.040, 0\> \quad [C_1=0.040 :: C_2=0.040] \{\} \)

\(<a_1, 0>\)
IRTM: System Model

Simulation

State: \(\langle 0.040, 1 \rangle \left[ C_1=0 :: C_2=0.040 \right] \{ \}

\(\langle a_1, 0 \rangle\)
IRTM: System Model

State: \( \langle 0.040, 1 \rangle \quad \left[ \begin{array}{c} c_1=0 \\ c_2=0.040 \end{array} \right] \} \)
IRTM: System Model

Simulation

State: \( \langle 0.040, 2 \rangle \left[ C_1=0 :: C_2=0.040 \right] \{ \}

\( \langle a_1, 0 \rangle \times \langle a_2, 0.040 \rangle \)
IRTM: System Model

Simulation

State: \(\langle 0.040, 3 \rangle \left[ c_2 = 0.040 \right] \{ \} \)
IRTM: System Model

State: \(\langle 0, 2 \rangle \left[ c_1=0 : c_2=0 \right] \{e_1, e_2\}\)
State: \( \langle 0, 2 \rangle \left[ \begin{array}{c} C_1 = 0 \end{array} \right] \left[ \begin{array}{c} C_2 = 0 \end{array} \right] \{ e_1, e_2 \} \)
IRTM: System Model

Simulation

State: $\langle 0, 3 \rangle \left[ C_1=0 : C_2=0 \right] \{ e_1, e_2 \}$
**IRTM: System Model**

State: \( \langle 0, 3 \rangle \) \[ C_1 = 0 :: C_2 = 0 \] \{ e_1, e_2 \}

Send \( \min(a_1, a_3) \)

Simulation

\[ \begin{align*}
    e_1? & \rightarrow a_1! \\
    & \xrightarrow{0.125\; ms} a_2! \\
    e_2? & \rightarrow a_3! \\
    & \xrightarrow{93\; ms} a_4! \\
    & \xrightarrow{93\; ms} a_5! \\
\end{align*} \]
IRTM: Environment Model

\[ e_1 \rightarrow 0.5 \text{ mu} \rightarrow e_2 \rightarrow 0.5 \text{ mu} \rightarrow e_3 \rightarrow 0.5 \text{ mu} \]
Non-Determinism: missed events

Interpretation

Non-determinism

missed events

TA aspects
Non-Determinism: missed events

Interpreter: 

\( \mathcal{E} \)

Non-Determinism: missed events

Diagram: 

- \( e_1! \) to \( e_2! \) with label \( 0.5 \text{ mu} \)
- \( e_2! \) to \( e_3! \) with label \( 0.5 \text{ mu} \)
- \( e_3! \) to \( e_3! \) with label \( 0.5 \text{ mu} \)

TA aspects: 

- Missed events

Interpretation: 

Non-determinism
Non-Determinism: duration variation

duration bounds

Interpretation

Non-determinism

TA aspects
Outline

1. Objectives

2. Interactive Real-Time Model

3. Testing Framework
Contribution

Publications

- **Burloiu, Cont, Poncelet.** A visual framework for dynamic mixed music notation. Journal of New Music Research (JNMR, 2016).

Developments

- Application to Antescofo + Regression tests
- Front-end compiler of Antescofo DSL (C++, 13.000 loc).
- Antescofo adaptors (C++).
- ~ 20 Scripts for test execution (Perl).
- Conformance and trace manager (C++, 4.000 loc).
- Virtual Machine (C++, 3.000 loc).
Perspectives

**Applications:**
- Visual tool for improving framework uses
  - A debug environment with Ascograph
- Application to other IMS (or timed-cyber systems)

**Interactive Real-Time Model:**
- Translate IRTM into Hybrid Automata
  - add constraints on tempo
- Translate IRTM into Stochastic Automata
  - Improve the input generation

**Testing Framework:**
- Improve Fuzz Testing
  - White-fuzzing Guided-Random (DART)
- Specify a concrete specification language
  - Based on a *Given-When-Then* like paradigm (Gherkin)

**Other Applications:**
- Static Analysis of Mixed Score
- Verification of properties
Testing Framework Overview

Mixed Score → Models

Construction: from high level to model

Model-Based Testing: from model to verdict

Offline Approach

Verdict

Online Approach

C. Poncelet, F. Jaquemard.
Translation: from IRTM into TA

Build
Simulate
Verify

Covering Input Generation

CoVer

- Covering generation
- Existing Tools
- Translation into Timed Automata
- Generates lowest durations

covering queries
observers
Location/Transition/Path

C. Poncelet, F. Jaquemard.
Test Methods for Score-Based Interactive Music Systems.
ICMC - SMS, 2014.

Blom, Hessel, Jonsson, Peterson.
Specifying and Generating Test Cases Using Observer Automata.
FATES’04.
Covering Input Generation

CoVer

- Covering generation
- Existing Tools
- Translation into Timed Automata
- Generates lowest durations

Covering queries
Observers
Location/Transition/Path

Covering Input Generation

CoVer

- Covering generation
- Existing Tools
- Translation into Timed Automata
- Generates lowest durations

covering queries

Location/Transition/Path

observers

Blom, Hessel, Jonsson, Peterson. 
Specifying and Generating Test Cases Using Observer Automata. 
FATES’04.
Covering Input Generation

- Covering generation
- Existing Tools
- Translation into Timed Automata
- Generates lowest durations

Fuzz Generation

- Musically relevant
- No translation
- No coverage guarantee

Random

Time Functions (TIF)

f

\[ f(t) = \begin{cases} 1 & t < T_1 \\ 0 & t \geq T_1 \end{cases} \]

\[ g(t) = \begin{cases} 1 - |t - T_2| & |t - T_2| < 1 \\ 0 & \text{otherwise} \end{cases} \]

\[ h(t) = \begin{cases} 1 & t < T_3 \\ 0 & t \geq T_3 \end{cases} \]

a) Tempo

b) Time-shift

c) Time-map

Interpretation = local shift and global tempo changes

Henkjan Honing.
Execution
Timed Conformance

Definition:

Timed conformance:

Set inclusion of real timed output traces into the expected timed output traces

Expected Trace:

\[ \langle a_1, 0 \times a_2, 0.040 \times a_3, 0.080 \rangle \]

Real Trace:

\[ \langle a_1, 0 \times a_2, 0.040 \times a_3, 0.080 \rangle \]
Timed Conformance

Definition:
Timed conformance:
Set inclusion of real timed output traces into the expected timed output traces

Expected Trace:
\( \langle a_1, 0 \rangle \times a_2, 0.040 \times a_3, 0.080 \rangle \)

Real Trace:
\( \langle a_1, 0 \rangle \quad \langle a_3, 0.080 \rangle \)
Timed Conformance

**Definition:**

Timed conformance:

Set inclusion of real timed output traces into the expected timed output traces

- **Expected Trace:** $\langle a_1, 0 \rangle$  $\langle a_3, 0.080 \rangle$

- **Real Trace:** $\langle a_1, 0 \rangle \times a_2, 0.040 \times a_3, 0.080 \rangle$

- **Unexpected actions**
Timed Conformance

**Definition:**

Timed conformance:

Set inclusion of real timed output traces into the expected timed output traces

\[ \langle a_1, 0 \times a_2, 0.040 \times a_3, 0.080 \rangle \]

\[ \langle a_1, 0 \times a_2, 0.040 \times a_3, 0.090 \rangle \]

Delta > $\varepsilon$
Testing Framework Overview

Online Approach

- No translation
- No coverage guarantee

(Simulation) Compute Expected Outputs
On-the-fly Inputs Generation
(Execution) Compute Real Outputs

Model = bytecode

Virtual Machine
Generator

On-the-fly Comparison

Implementation Under Test
1. Objectives

2. Interactive Real-Time Model

3. Testing Framework

4. Application to Antescofo
Application to Antescofo
Antescofo

Listening machine

tempo
pos.

Reactive engine
Offline Approaches application

Mixed Score (DSL Antescofo)

Model

Inputs Generation

1

(Simulation)
Compute Expected Outputs

tin

tref

2

(Timed conformance)

3

(Execution)
Compute Real Outputs

tout

Construction: from high level to model

4

= =

Timed conformance

from high level to model
Construction:
from high level to model

Domain Specific Language
Antescofo

Interactive Real-Time Model

Automatic

Inference Rules

FSM Parts & Connectors + Operators

Construction

\[
\begin{align*}
1 & : E \to a \to E' \quad 1 \\
2 & : \overline{E} \to \overline{a} \to \overline{E'} \quad 2 \\
3 & : E_e \to \overline{a} \to E'_e \quad 3 \\
4 & : \overline{E}_e \to \overline{a} \to \overline{E}_e' \quad 4 \\
\end{align*}
\]

\[
\begin{align*}
\text{Inference Rules:} & \\
\text{FSM Parts & Connectors + Operators:} & \\
: ms \vdash_{\text{env}} M_{\text{env}} : ms \vdash_{\text{proxy}} P : ms \vdash_{\text{sys}} A \\
: ms \vdash_{\text{all}} M_{\text{env}} \| P \| A \\
\end{align*}
\]
Antescofo Execution

\[ \langle e_1, 0, 120 \rangle \times \langle e_2, 0.5, 120 \rangle \times \langle e_3, 1, 120 \rangle \]

\[ \langle e, t, p \rangle \]
Antescofo Execution

\( \langle e_1, 0, \_ \rangle \times \langle e_2, 0.5, \_ \rangle \times \langle e_3, 1, \_ \rangle \)

\( \langle e, t \rangle \)

black box
Verdict

```
start of tests
note0  \rightarrow  0.77 \rightarrow  note1_1.00 \rightarrow  0.385 \rightarrow  note2_1.50 \rightarrow  0.1925 \rightarrow  note3_1.75 \rightarrow  0

<table>
<thead>
<tr>
<th>TimeStamp</th>
<th>Antescofo Trace</th>
<th>Expected Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>a0</td>
<td>0 [ 0]</td>
<td>a0 0 [ 0]</td>
</tr>
<tr>
<td>note0</td>
<td>0 [ 0]</td>
<td>e0 0 [ 0]</td>
</tr>
<tr>
<td>a1</td>
<td>0.5 [ 0.5]</td>
<td>a1 0.5 [ 0.5]</td>
</tr>
<tr>
<td>a2</td>
<td>0.77 [0.77]</td>
<td>a2 0.77 [0.77]</td>
</tr>
<tr>
<td>note1_1.00</td>
<td>0.77 [0.824]</td>
<td>e1 0.77 [0.77]</td>
</tr>
<tr>
<td>a3</td>
<td>0.887 [0.95]</td>
<td>a3 0.887 [1.02]</td>
</tr>
<tr>
<td>note2_1.50</td>
<td>0.95047 [1.21]</td>
<td>e2 0.95047 [1.16]</td>
</tr>
<tr>
<td>a4</td>
<td>0.99 [1.29]</td>
<td>a4 0.95 [1.16]</td>
</tr>
<tr>
<td>a5</td>
<td>1.02 [1.35]</td>
<td>a5 0.985 [1.25]</td>
</tr>
<tr>
<td>a6</td>
<td>1.02 [1.35]</td>
<td>a6 1.02 [1.35]</td>
</tr>
<tr>
<td>note3_1.75</td>
<td>1.0174 [1.4]</td>
<td>e3 1.0174 [1.35]</td>
</tr>
<tr>
<td>note4_1.75</td>
<td>1.0174 [1.4]</td>
<td>e4 1.0174 [1.35]</td>
</tr>
<tr>
<td>a7</td>
<td>1.28 [2.3]</td>
<td>a7 1.28 [2.31]</td>
</tr>
<tr>
<td>END</td>
<td>1.2831 [2.38]</td>
<td>e5 1.2831 [2.31]</td>
</tr>
</tbody>
</table>

Error :: Test KO
```

78
Experiments

- Coverage (% locations)
  - 0%
  - 5%
  - 50%
- Consecutive misses
  - 0
  - 1
  - 3
  - 5
  - 7
- Duration (seconds)
  - 0
  - 100
  - 200
  - 300
  - 400

Approach Offline: CoVer

Benchmark

- Consecutive misses (% allowed duration variation)
- Translation into Timed Automata
- No musically relevant

• Covering generation
• Existing Tools
Experiments

Sonata in F major
Georg Friedrich Händel

Measures 5
10s
25 events
84 actions

Measures 8
16s
48 events
185 actions

Measures 10
20s
74 events
264 actions

Measures 15
30s
122 events
444 actions

Measures 40
80s
360 events
1218 actions

Approach Offline: CoVer

misses - k

coverage (% locations)

0-00 3-10 7-25 consecutive misses - % allowed duration variation

0
25
50
75
100

Measures
5 8 10 15 40

0
5 8 10 15 40

Nb generated traces

0
25
50
75
100

measures
Experiments

Sonata in F major
Georg Friedrich Händel

Approach Offline: CoVer

• Not scalable
Experiments

Sonata in F major
Georg Friedrich Händel

Approach Offline: Fuzz

10 traces

40 measures

---

Musically relevant
No translation
No coverage guarantee

consecutive misses - % allowed duration variation

Measures 40
80s
360 events
1218 actions
Experiments

- Sonata in F major
- Georg Friedrich Händel

**Approach Online**

**Experiments**

- **misses - k**

10 traces

<table>
<thead>
<tr>
<th>Nb. tins</th>
<th>coverage (%) locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>58</td>
</tr>
<tr>
<td>50</td>
<td>61</td>
</tr>
<tr>
<td>100</td>
<td>62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nb. tins</th>
<th>duration (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>75</td>
</tr>
<tr>
<td>50</td>
<td>150</td>
</tr>
<tr>
<td>100</td>
<td>300</td>
</tr>
</tbody>
</table>

- Musically relevant
- No translation
- No coverage guarantee
- Faster
Appendices

State: $\langle 0, 0 \rangle \left[ c_1=0 :: c_2=0 \right] \{ \}$
Appendices

State: \[ \langle 0, 0 \rangle \left[ C_1 = 0 :: C_2 = 0 \right] \{ \} \]
Appendices

State: $\langle 0, 1 \rangle \left[ c_1 = 0 :: c_2 = 0 \right] \{ a \}$
State: $\langle 0, 2 \rangle \left[ \begin{array}{c} c_1 = 0 \\ c_2 = 0 \end{array} \right] \{ a, b \}$
Appendices

State: $\langle 0, 3 \rangle \left[ C_2=0 \right] \{ a, b \}$
Appendices

State: $\langle 0, 4 \rangle \left[ \begin{array}{c} C_2=0 \end{array} \right] \{ a, b \}$
State: $\langle 0, 5 \rangle \left[ C_2 = 0 \right] \{ a, b \}$