Improving Translation of Live Sequence Charts to Temporal Logic

Rahul Kumar  Eric Mercer  Annette Bunker

AVOCS 2007
Trends: SoC, Multi-agent Systems
Traditional Testing
What do IP Core vendors and consumers need?

Interfaces need to be defined and their implementations verified.
Scenario Based Specifications

- R.Z. ITU-T 120: Message Sequence Charts
- Damm et. al.: Live Sequence Charts
- Bunker et. al.: Protocol Live Sequence Charts
What do we have so far?

- Büchi automata
- Temporal Logics
  - CTL*, CTL, LTL
- Graphical Languages
  - LSCs, UML, PLSCs

Increasing level of use

Increase in verification support
Verification Using Scenarios

LSC → LTL_A → LTL_B → LTL_C → Model Checker → ?

Bunker et. al. 2002

Theoretically correct
Difficult to use

Easy to use
Formally incomplete
Verification Using Scenarios

- Theoretically correct
- Difficult to use

Bunker et. al. 2002
- Easy to use
- Formally incomplete

LSC → LTL → Model Checker

≈
Verification Using Scenarios

- Theoretically correct
- Difficult to use

Westphal, Toben et. al. 2006

Bunker et. al. 2002
- Easy to use
- Formally incomplete

LSC → Automata → Model Checker
- LTL

Automated Verification of Critical Systems 2007
Computer Science Department
Brigham Young University
Provo, UT 84606, USA
Problems & Solutions

Bridge the gap between graphical specifications and verification methodologies
Live Sequence Charts

- Initiator
- Target
- Processes
- Pre-chart
- Messages
- Universal main chart

Symbols:
- P
- A
- L
- S
Kugler’s Approach

\[ G( \Theta_{\text{pre}} \Rightarrow \Theta_{\text{main}} ) \]
Kugler’s Approach

\[ G( P \Rightarrow \Theta_{\text{main}} ) \]
Kugler’s Approach - Ordering

G( P ⇒ 

(¬L UA) ∧ 
(¬S UA) ∧ 
(¬T UA) ∧ 
(¬S UL) ∧ 
(¬T UL) ∧ 
(¬T US) ∧ 
(F T))
Kugler’s Approach - Uniqueness

G( P ⇒ ¬χ_{A,L} ∧ ¬χ_{A,S} ∧ ¬χ_{A,T} ∧ 
   ¬χ_{L,S} ∧ ¬χ_{L,T} ∧
   ¬χ_{S,T} )

¬χ_{a,b^+} : (¬b ∧ ¬a) U (a ∧ X((¬b ∧ ¬a) U a))
Kugler’s Approach - Total

\[ G( P \Rightarrow (\neg L U A) \land \neg \chi_{A,L} \land \neg \chi_{A,S} \land \neg \chi_{A,T} \land \neg S U A) \land \neg \chi_{L,S} \land \neg \chi_{L,T} \land \neg \chi_{S,T} \land \neg S U L) \land \neg T U A) \land \neg T U L) \land \neg T U S) \land (F T)) \]
Kugler’s Approach - Drawbacks

• Very large formulas for small charts
• LTL to automata fails for large charts
• Verification fails for larger charts/models
• Limited set of LSC constructs translated
Reductions: Until Transitivity
Reductions: Using Until Reduction

\[ G( \text{pre} \Rightarrow (\neg L \cup A) \land (\neg S \cup A) \land (\neg T \cup A) \land (\neg S \cup L) \land (\neg T \cup L) \land (\neg T \cup S) \land (F T)) \]
Reductions: Uniqueness

Using Coverage to Specify Uniqueness

- Red does not occur again before Blue
- Red does not occur again before Yellow
- Red does not occur again before Green

Established order of events: Red, Yellow, Green, Blue

∧
Using Uniqueness Reduction

\[ G(\text{pre} \implies \chi_{A,L} \land \chi_{A,S} \land \chi_{A,T} \land \neg \chi_{L,S} \land \neg \chi_{L,T} \land \neg \chi_{S,T}) \]
Additional Constructs

**Existing charts**

**Conditions**
- Asynchronous messages
- Co-region

**Initiator**

```
cmdval == 0
```

**Target**

```
START
U
T
E
REQ
ACK
```
Co-regions

\[(\neg c \cup a) \land \neg c \cup b \land F c\]

Force a, b to occur but in any order
Asynchronous Messages

Single letter per message

Describe send and receive events
Asynchronous Messages

(¬b? U b!) ∧ (¬a? U a!) ∧ (assumption)
(¬b? U a!) ∧ (¬a? U b!) ∧
(¬c? U a?) ∧ (¬c! U b?) ∧
F c? ∧ F c!
We only care about \textit{bot} here. Open at other locations else!
Conditions

Bonded Conditions

Non-bonded Conditions

Translation undecided
Analysis

Ordering

$n$-k properties to order first $n$ messages before $k$ messages

$k$ properties for enforcing occurrence of final $k$ messages
Analysis

Uniqueness

\( k \) properties for each of the \( n \) messages

\( (k-1) \) properties for each of the \( k \) messages

\( n^*k \)

Worst case chart
Analysis

For chart of size $n \& 1$ maximal message:
Ordering: $n$ properties
Uniqueness: $n$ properties

\[ n-1 \]
Analysis

Multiple maximal messages \((m)\):
Ordering: \(n\)
Uniqueness: \(n \times m\)

As \(m \rightarrow n\), formula becomes quadratic
Theoretical Results

- Ordering in linear properties
- Uniqueness in sub-quadratic properties
- Translation at most as opposed to at least quadratic
- Additional constructs
  - Existential charts
  - Co-regions
  - Asynchronous messages
  - Invariants and bonded conditions
Experiments - Specifications

A3 specification

P₀  P₁  P₂  P₃  P₃

a₀  a₁  a₂  a₃  a₄  a₅  a₆  a₇  a₈

Other specifications with 5-7 messages
Experiments - Models and Verifiers

• Models
  – Promela models with simple message passing
  – Puzzle models followed by messages
  – “_e” models contain errors in main chart
• SPIN and NuSMV for model checking
• LTL2BA: explicit state automata generation
### Empirical Results - LTL2BA

<table>
<thead>
<tr>
<th>Specification</th>
<th>Messages</th>
<th>Kugler’s Translation</th>
<th>Improved Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Messages</td>
<td>Maximal Messages</td>
<td>Size</td>
</tr>
<tr>
<td>SpecA</td>
<td>5</td>
<td>1</td>
<td>209</td>
</tr>
<tr>
<td>SpecB</td>
<td>5</td>
<td>2</td>
<td>175</td>
</tr>
<tr>
<td>SpecC</td>
<td>7</td>
<td>2</td>
<td>LTL2BA DNF</td>
</tr>
</tbody>
</table>
## Empirical Results - SPIN

<table>
<thead>
<tr>
<th>Specification</th>
<th>Model</th>
<th>Kugler’s Translation</th>
<th>Improved Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>States</td>
<td>Time (s)</td>
</tr>
<tr>
<td>SpecB</td>
<td>SysA</td>
<td>2612</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>SysA_e</td>
<td>2446</td>
<td>0.07</td>
</tr>
<tr>
<td>SpecC</td>
<td>SysB</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SysB_e</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>Soko</td>
<td>3847560</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>Soko_e</td>
<td>1479320</td>
<td>32</td>
</tr>
<tr>
<td>A3</td>
<td>Soko</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Soko_e</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
## Empirical Results - NuSMV

<table>
<thead>
<tr>
<th>Specification</th>
<th>Model</th>
<th>States</th>
<th>Kugler’s Time (s)</th>
<th>Improved Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A2</strong></td>
<td>bridge</td>
<td>76992</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Abp4</td>
<td>2236420</td>
<td>30</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Bridge_e</td>
<td>76992</td>
<td>29</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Abp4_e</td>
<td>2236420</td>
<td>76</td>
<td>27</td>
</tr>
<tr>
<td><strong>A3</strong></td>
<td>bridge</td>
<td>76992</td>
<td>22</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>abp4</td>
<td>2236420</td>
<td>59</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Bridge_e</td>
<td>76992</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Abp4_e</td>
<td>2236420</td>
<td>146</td>
<td>50</td>
</tr>
<tr>
<td><strong>A4</strong></td>
<td>bridge</td>
<td>76992</td>
<td>49</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>abp4</td>
<td>2236420</td>
<td>174</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Bridge_e</td>
<td>76992</td>
<td>132</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Abp4_e</td>
<td>2236420</td>
<td>337</td>
<td>67</td>
</tr>
<tr>
<td><strong>A5</strong></td>
<td>bridge</td>
<td>76992</td>
<td>175</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>abp4</td>
<td>2236420</td>
<td>555</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>Bridge_e</td>
<td>76992</td>
<td>509</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>Abp4_e</td>
<td>2236420</td>
<td>1271</td>
<td>131</td>
</tr>
</tbody>
</table>
Conclusions

• Reductions produce vast improvement
• Scalability still limited in explicit state
• Translating constructs can be difficult
• Is this translation minimal?
• What about LSC to automata?
LSC To Automata: Traditional Method
LSC To Automata: Reachability

Verify safety.
LSC To Automata: Liveness

Verify $AGAF(fair)$ to enforce progress
LSC to Automata: Conditions

How do you detect errors without introducing non-determinism?
Drawbacks

• Safety and progress checking performed in two separate runs
• Non-determinism because of conditions
• Undecided semantics of conditions
New Solution

Safety and Liveness in one run by verifying $EGEF(error)$
Conditions in New Solution

Placement of error state fixes the problem of non-determinism as well as detecting errors!
Advantages

• One shot verification using LSCs
• All constructs supported
• No non-determinism
• More error states means faster detection of errors in system
• Simple unwinding algorithm