Softwaretechnik / Software-Engineering

Lecture 09: Live Sequence Charts

2015-06-11

Prof. Dr. Andreas Podelski, Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:
- Scenarios and Anti-Scenarios
- User Stories, Use Cases, Use Case Diagrams
- LSC: abstract and concrete syntax

This Lecture:
- Educational Objectives: Capabilities for following tasks/questions.
  - Which are the cuts and firedsets of this LSC?
  - Construct the TBA of a given LSC body.
  - Given a set of LSCs, which scenario/anti-scenario/requirement is formalised by them?
  - Formalise this positive scenario/anti-scenario/requirement using LSCs.

- Content:
  - Excursion: automata accepting infinite words
  - Cuts and Firedsets, automaton construction
  - existential LSCs, pre-charts, universal LSCs
  - Requirements Engineering: conclusions
Recall: LSC Body Syntax

LSC Body Example

- $L$: $l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}$, $l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}$, $l_{3,0} \prec l_{3,1} \prec l_{3,2}$, $l_{3,1} \prec l_{3,2}$, $l_{2,2} \prec l_{1,2} \prec l_{1,3} \prec l_{2,3}$, $l_{2,3} \prec l_{1,4} \prec l_{3,2} \sim l_{3,1}$.
- $I = \{(l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}), (l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}), (l_{3,0}, l_{3,1}, l_{3,2})\}$.
- $Msg = \{(l_{1,1}, A, l_{1,2}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{3,2}), (l_{3,2}, E, l_{3,4})\}$.
- $Cond = \{(l_{2,2}, c_2 \land c_3)\}$.
- $LocInv = \{(l_{1,1}, \circ, l_{1,2}, \bullet)\}$.

LSC Semantics
Recall: decision tables

By the standard semantics, a decision table $T$ is software, 
$\{ \sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots \} \}$ is a set of computation paths.

Recall: Decision tables as software specification:

We want the same for LSCs.

We will give a procedure to construct for each LSC $L$ an automaton $B(L)$. The language (or semantics) of $L$ is the set of comp. paths accepted by $B(L)$. Thus an LSC is also software.

Problem: computation paths may be infinite $\rightarrow$ Büchi acceptance.

Excursion: Symbolic Büchi Automata
Symbolic Büchi Automata

**Definition.** A **Symbolic Büchi Automaton** (TBA) is a tuple

\[ B = (C, Q, q_{ini}, \rightarrow, Q_F) \]

where

- \( C \) is a set of atomic propositions,
- \( Q \) is a finite set of **states**,  
- \( q_{ini} \in Q \) is the initial state,
- \( \rightarrow \subseteq Q \times \Phi(C) \times Q \) is the finite **transition relation**.

Each transition \((q, \psi, q') \in \rightarrow\) from state \(q\) to state \(q'\) is labelled with a formula \(\psi \in \Phi(C)\).

- \(Q_F \subseteq Q\) is the set of **fair** (or accepting) states.
**Run of TBA**

**Definition.** Let $\mathcal{B} = (C, Q, q_{\text{ini}}, \rightarrow, Q_F)$ be a TBA and

$$w = \sigma_1, \sigma_2, \sigma_3, \ldots \in (C \rightarrow \mathcal{B})^\omega$$

an infinite word, each letter is a valuation of $C$. An infinite sequence

$$q = q_0, q_1, q_2, \ldots \in Q^\omega$$

of states is called **run** of $\mathcal{B}$ over $w$ if and only if

- $q_0 = q_{\text{ini}},$
- for each $i \in N_0$ there is a transition $(q_i, \psi_i, q_{i+1}) \in \rightarrow$ s.t. $\sigma_i \models \psi_i$.

**Example:**

$\omega = (x=0)(x\neq 4)(x=4)(x\neq 8)(x\neq 12)\ldots$

$B_{\text{sym}}$: even $(x) \Sigma = (\{x\} \rightarrow N)$

$\mathcal{L} = \{q_1, q_1, q_2, q_1, q_1, q_2, \ldots\}$

---

**The Language of a TBA**

**Definition.** We say TBA $\mathcal{B} = (C, Q, q_{\text{ini}}, \rightarrow, Q_F)$ **accepts** the word

$$w = (\sigma_i)_{i \in N_0} \in (C \rightarrow \mathcal{B})^\omega$$

if and only if $\mathcal{B}$ has a run

$$q = (q_i)_{i \in N_0}$$

over $w$ such that fair (or accepting) states are **visited infinitely often** by $q$, i.e., such that

$$\forall i \in N_0 \exists j > i : q_j \in Q_F.$$

We call the set $\text{Lang}(\mathcal{B}) \subseteq (C \rightarrow \mathcal{B})^\omega$ of words that are accepted by $\mathcal{B}$ the **language** of $\mathcal{B}$. 

---
run: \( q_0, q_1, q_2, \ldots \in Q^\omega \) s.t. \( \sigma_i \models \psi_i, \ i \in \mathbb{N}_0 \).

LSC Semantics: TBA Construction
Definition. Let \((L, \preceq, \sim), I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta)\) be an LSC body.

A non-empty set \(\emptyset \neq C \subseteq L\) is called a cut of the LSC body iff \(C\)

- is downward closed, i.e.
  \[\forall l, l' \in L \cdot l' \in C \land l \preceq l' \implies l \in C,\]

- is closed under simultaneity, i.e.
  \[\forall l, l' \in L \cdot l' \in C \land l \sim l' \implies l \in C,\]

- comprises at least one location per instance line, i.e.
  \[\forall I \in I \cdot C \cap I \neq \emptyset.\]

The temperature function is extended to cuts as follows:

\[\Theta(C) = \begin{cases} 
  \text{hot}, & \text{if } \exists l \in C \cdot (\exists l' \in C \cdot l \prec l') \land \Theta(l) = \text{hot} \\
  \text{cold}, & \text{otherwise}
\end{cases}\]

that is, \(C\) is hot if and only if at least one of its maximal elements is hot.

Cut Examples

\(\emptyset \neq C \subseteq L\) — downward closed — simultaneity closed — at least one loc. per instance line
A Successor Relation on Cuts

The partial order “⪯” and the simultaneity relation “∼” of locations induce a **direct successor relation** on cuts of \( \mathcal{L} \) as follows:

**Definition.** Let \( C \subseteq \mathcal{L} \) be a cut of LSC body \( (\mathcal{L}, \preceq, \sim), \mathcal{I}, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta) \). A set \( \emptyset \neq \mathcal{F} \subseteq \mathcal{L} \) is called **fired-set** \( \mathcal{F} \) of \( C \) if and only if

- \( C \cap \mathcal{F} = \emptyset \) and \( C \cup \mathcal{F} \) is a cut, i.e. \( \mathcal{F} \) is closed under simultaneity,
- all locations in \( \mathcal{F} \) are **direct \(-\)-successors** of the front of \( C \), i.e.
  \[ \forall l \in \mathcal{F} \exists l' \in C \land l' \prec l \land (\exists l'' \in C \land l' \prec l'') \],
- locations in \( \mathcal{F} \), that lie on the same instance line, are **pairwise unordered**, i.e.
  \[ \forall l \neq l' \in \mathcal{F} \land (\exists I \in \mathcal{I} \land \{l, l'\} \subseteq I) \Rightarrow l \not\preceq l' \land l' \not\preceq l \],
- for each asynchronous message reception in \( \mathcal{F} \), the corresponding **sending** is already in \( C \),
  \[ \forall (l, E, l') \in \text{Msg} \land l' \in \mathcal{F} \Rightarrow l \in C \].

The cut \( C' = C \cup \mathcal{F} \) is called **direct successor of \( C \) via \( \mathcal{F} \)**, denoted by \( C \leadsto_{\mathcal{F}} C' \).
Successor Cut Example

\[ C \cap F = \emptyset \quad \text{only direct } \prec \text{-successors} \quad \text{same instance line on front} \]
\[ \text{pairwise unordered} \quad \text{sending of asynchronous reception already in} \]
Recall: The TBA $B(L)$ of LSC $L$ is $(C, Q, q_{ini}, \rightarrow, Q_F)$ with

- $Q$ is the set of cuts of $L$, $q_{ini}$ is the instance heads cut,
- $C = C \cup \mathcal{E}_F$, where $\mathcal{E}_F = \{ E! , E? \mid E \in \mathcal{E} \}$,
- $\rightarrow$ consists of loops, progress transitions (from $\sim_F$), and legal exits (cold cond./local inv.),
- $Q_F = \{ C \in Q \mid \Theta(C) = \text{cold} \lor C = \mathcal{L} \}$ is the set of cold cuts and the maximal cut.

So in the following, we "only" need to construct the transitions’ labels:

$$\rightarrow = \{ (q, \psi_{loop}(q), q) \mid q \in Q \} \cup \{ (q, \psi_{prop}(q, q'), q') \mid q \sim_F q' \} \cup \{ (q, \psi_{exit}(q), \mathcal{L}) \mid q \in Q \}$$
So in the following, we "only" need to construct the transitions' labels:

\[ \rightarrow = \{(q, \psi_{\text{loop}}(q), q) \mid q \in Q\} \cup \{(q, \psi_{\text{prog}}(q, q'), q') \mid q \rightarrow_{F} q'\} \cup \{(q, \psi_{\text{exist}}(q), L) \mid q \in Q\} \]

\[ \psi_{\text{loop}}(q) = \psi_{\text{Msg}}(q) \land \psi_{\text{LocInv}}(q) \land \psi_{\text{LocInv}}(q) \]

\[ \psi_{\text{prog}}(q, q_n) = \psi_{\text{Msg}}(q) \land \psi_{\text{LocInv}}(q, q_n) \land \psi_{\text{LocInv}}(q, q_n) \]

\[ \psi_{\text{exit}}(q) = (\psi_{\text{Msg}}(q) \land \neg \psi_{\text{LocInv}}(q, q_n)) \lor \bigcup_{1 \leq i \leq n} (\psi_{\text{Msg}}(q, q_i) \land \neg \psi_{\text{LocInv}}(q, q_i)) \]

**Loop Condition**

- \( \psi_{\text{Msg}}(q) = \bigwedge_{1 \leq i \leq n} \psi_{\text{Msg}}(q, q_i) \land \left(\text{strict} \implies \bigwedge_{\psi \in E, j \in \text{Msg}(L) \land \neg \psi} M_{j}: q \not\in F\right) \)
- \( \psi_{\text{LocInv}}(q) = \bigwedge_{t = \langle l, l_0, l_1, l_2 \rangle \in \text{LocInv}, \Theta(t) = \theta} \ell \text{ active at } q \)

A location \( l \) is called **front location** of cut \( C \) if and only if \( \nexists l' \in L : l \prec l' \).

Local invariant \( (l_0, l_0, \phi, l_1, l_2) \) is **active** at cut (1) \( q \) if and only if \( l_0 \preceq l \preceq l_1 \) for some front location \( l \) of cut (1) \( q \).

- \( \text{Msg}(F) = \{E! \mid (l, E, l') \in \text{Msg}, \ l \in F\} \cup \{E? \mid (l, E, l') \in \text{Msg}, \ l' \in F\} \)
- \( \text{Msg}(F_1, \ldots, F_n) = \bigcup_{1 \leq i \leq n} \text{Msg}(F_i) \)
Progress Condition

\[ \psi_{\text{hot}}(q_0, q_i) = \psi_{\text{Mag}}(q_0, q_i) \land \psi_{\text{Cond}}(q_0, q_i) \land \psi_{\text{LocInv}}(q_i) \]

- \( \psi_{\text{Mag}}(q_0, q_i) = \bigwedge_{\psi \in \text{Mag}(q_i \setminus q_0)} \psi \land \bigwedge_{\psi \notin \text{Mag}(q_i \setminus q_0)} \neg \psi \) (strict)

- \( \psi_{\text{Cond}}(q_0, q_i) = \bigwedge_{\gamma = (L, \phi) \in \text{Cond}, \Theta(\gamma) = \theta, L \cap (q_i \setminus q_0) \neq \emptyset} \phi \)

- \( \psi_{\text{LocInv}}(q_0, q_i) = \bigwedge_{\lambda = (l, \iota, \phi, l') \in \text{LocInv}, \Theta(\lambda) = \theta} \phi \)

Local invariant \((l_0, l_0, \phi, l_1, l_1)\) is \textit{\(\bullet\)-active} at \(q\) if and only if

- \( l_0 < l < l_1 \), or
- \( l = l_0 \land l_0 = \bullet \), or
- \( l = l_1 \land l_1 = \bullet \)

for some front location \(l\) of cut (!) \(q\).
A full LSC $\mathcal{L} = (((L, \preceq, \sim), I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta), ac_0, am, \Theta_\mathcal{L})$ consist of
- body $((L, \preceq, \sim), I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta)$,
- activation condition $ac_0 \in \Phi(C)$, strictness flag strict (otherwise called permissive)
- activation mode $am \in \{\text{initial}, \text{invariant}\}$,
- chart mode existential ($\Theta_\mathcal{L} =$ cold) or universal ($\Theta_\mathcal{L} =$ hot).

Concrete syntax:

An LSC $W \subseteq (C \rightarrow B)^\omega$ is accepted by $\mathcal{L}$ if and only if

<table>
<thead>
<tr>
<th>$\Theta_\mathcal{L}$</th>
<th>$am = \text{initial}$</th>
<th>$am = \text{invariant}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{p}_0$</td>
<td>$\exists w \in W \cdot w^0 \models ac \land w^0 \models \psi_{\text{cond}}(\emptyset, C_0) \land w/1 \in \text{Lang}(B(\mathcal{L}))$</td>
<td>$\exists w \in W \cdot \exists k \in \mathbb{N}<em>0 \cdot w^k \models ac \land w^k \models \psi</em>{\text{cond}}(\emptyset, C_0) \land w/k+1 \in \text{Lang}(B(\mathcal{L}))$</td>
</tr>
<tr>
<td>$\text{hot}$</td>
<td>$\forall w \in W \cdot w^0 \models ac \implies w^0 \models \psi_{\text{cond}}(\emptyset, C_0) \land w/k+1 \in \text{Lang}(B(\mathcal{L}))$</td>
<td>$\forall w \in W \cdot \forall k \in \mathbb{N}<em>0 \cdot w^k \models ac \implies w^k \models \psi</em>{\text{cond}}(\emptyset, C_0) \land w/k+1 \in \text{Lang}(B(\mathcal{L}))$</td>
</tr>
</tbody>
</table>

where $ac = ac_0 \land \psi_{\text{cond}}(\emptyset, C_0) \land \psi_{\text{Msg}}(\emptyset, C_0)$; $C_0$ is the minimal (or instance heads) cut.
References