Software Design, Modelling and Analysis in UML

Lecture 18: Live Sequence Charts II

2016-01-28

Prof. Dr. Andreas Podelski, Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:
- Rhapsody code generation
- Interactions: Live Sequence Charts
- LSC syntax

This Lecture:
- Educational Objectives: Capabilities for following tasks/questions.
  - How is the semantics of LSCs constructed?
  - What is a cut, fired-set, etc.?
  - Construct the TBA for this LSC.
  - Give one example which (non-)trivially satisfies this LSC.

- Content:
  - Symbolic Automata
  - Firedset, Cut
  - Automaton construction
  - Transition annotations
LSC Body: Abstract Syntax

Let $\Theta = \{\text{hot, cold}\}$. An LSC body is a tuple

$$(I, (L, \preceq), \sim, \mathcal{F}, \text{Msg}, \text{Cond}, \text{LocInv})$$

- $I$ is a finite set of instance lines,
- $(L, \preceq)$ is a finite, non-empty, partially ordered set of locations; each $l \in L$ is associated with a temperature $\theta(l) \in \Theta$ and an instance line $i_l \in I$,
- $\sim \subseteq L \times L$ is an equivalence relation on locations, the simultaneity relation,
- $\mathcal{F} = (\mathcal{F}, \mathcal{E}, V, \text{atr}, \delta')$ is a signature,
- $\text{Msg} \subseteq L \times E \times L$ is a set of asynchronous messages with $(l, b, l') \in \text{Msg}$ only if $l \preceq l'$, i.e., instantaneous messages — if $l \sim l'$ could be mapped to method/operation calls.
- $\text{Cond} \subseteq (2^L \setminus \emptyset) \times \mathcal{E} \times \Theta$ is a set of conditions where $\mathcal{E}$ are OCL expressions over $W = I \cup \{\text{self}\}$ with $(L, \mathcal{E}, \Theta) \in \text{Cond}$ only if $l \sim l'$ for all $l, l' \in L$,
- $\text{LocInv} \subseteq L \times \{\circ, \bullet\} \times \mathcal{E} \times \Theta \times L \times \{\circ, \bullet\}$ is a set of local invariants.
Well-Formedness

**Bondedness/no floating conditions:** (could be relaxed a little if we wanted to)

- For each location $l \in \mathcal{L}$, if $l$ is the location of
  - a *condition*, i.e. $\exists (L, expr, \theta) \in \text{Cond} : l \in L$, or
  - a *local invariant*, i.e. $\exists (l_1, i_1, expr, \theta, l_2, i_2) \in \text{LocInv} \in \{l_1, l_2\}$, or

then there is a location $l'$ *equivalent* to $l$, i.e. $l \sim l'$, which is the location of

- an *instance head*, i.e. $l'$ is minimal wrt. $\preceq$, or
- a *message*, i.e.

$$\exists (l_1, b, l_2) \in \text{Msg} : l \in \{l_1, l_2\}.$$

**Note:** if messages in a chart are *cyclic*, then there doesn’t exist a partial order (so such charts *don’t even have* an abstract syntax).

---

*Live Sequence Charts — Semantics*
**TBA-based Semantics of LSCs**

**Plan:**

- Given an LSC $L$ with body $(I, (\mathcal{P}, \leq), \sim, \mathcal{F}, \text{Msg}, \text{Cond}, \text{LocInv})$,
  - construct a TBA $B_L$, and
  - define language $L(L)$ of $L$ in terms of $L(B_L)$, in particular taking activation condition and activation mode into account.
- Then $\mathcal{M} \models L$ (universal) if and only if $L(M) \subseteq L(L)$.
  And $\mathcal{M} \models L$ (existential) if and only if $L(M) \cap L(L) \neq \emptyset$.

**Excursion: Büchi Automata**
Symbolic Büchi Automata

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

$$B = (\text{Expr}_B(X), X, Q, q_{\text{ini}}, \rightarrow, Q_{\text{F}})$$

where

- $X$ is a set of logical variables,
- $\text{Expr}_B(X)$ is a set of Boolean expressions over $X$,
- $Q$ is a finite set of states,
- $q_{\text{ini}} \in Q$ is the initial state,
- $\rightarrow \subseteq Q \times \text{Expr}_B(X) \times Q$ is the transition relation. Transitions $(q, \psi, q')$ from $q$ to $q'$ are labelled with an expression $\psi \in \text{Expr}_B(X)$.
- $Q_{\text{F}} \subseteq Q$ is the set of fair (or accepting) states.
**Definition.** Let $X$ be a set of logical variables and let $\text{Expr}_B(X)$ be a set of Boolean expressions over $X$.

A set $(\Sigma, \cdot \models \cdot)$ is called an **alphabet** for $\text{Expr}_B(X)$ if and only if

- for each $\sigma \in \Sigma$,
- for each expression $\text{expr} \in \text{Expr}_B$, and
- for each valuation $\beta : X \rightarrow \mathcal{D}(X)$ of logical variables to domain $\mathcal{D}(X)$,

  either $\sigma \models_\beta \text{expr}$ or $\sigma \not\models_\beta \text{expr}$.

An **infinite sequence**

$$w = (\sigma_i)_{i \in \mathbb{N}_0} \in \Sigma^\omega$$

over $(\Sigma, \cdot \models \cdot)$ is called **word** for $\text{Expr}_B(X)$.

---

**Run of TBA over Word**

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

$$w = \sigma_1, \sigma_2, \sigma_3, \ldots$$

a word for $\text{Expr}_B(X)$. An infinite sequence

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

is called **run of $B$ over $w$** under valuation $\beta : X \rightarrow \mathcal{D}(X)$ if and only if

- $q_0 = q_{\text{ini}}$,
- for each $i \in \mathbb{N}_0$ there is a transition $(q_i, \psi_i, q_{i+1}) \in \rightarrow$ such that $\sigma_i \models_\beta \psi_i$.

---

**Example:**

![TBA Diagram](image-url)
Definition.
We say TBA \( B = (Expr_B(X), X, Q, q_{ini}, \rightarrow, Q_F) \) accepts the word 
\( w = (\sigma_i)_{i \in \mathbb{N}_0} \in (Expr_B \to B)^\omega \) if and only if 
\( B \) has a run 
\( \rho = (q_i)_{i \in \mathbb{N}_0} \)
over \( w \) such that fair (or accepting) states are visited infinitely often by 
\( \rho \), i.e., such that 
\[ \forall i \in \mathbb{N}_0 \exists j > i : q_j \in Q_F. \]
We call the set \( \mathcal{L}(B) \subseteq (Expr_B \to B)^\omega \) of words that are accepted by \( B \) 
the language of \( B \).

Language of UML Model
**Words over Signature**

**Definition.** Let $\mathcal{F} = (\mathcal{F}, \mathcal{C}, V, atr, \mathcal{E})$ be a signature and $\mathcal{D}$ a structure of $\mathcal{F}$. A **word** over $\mathcal{F}$ and $\mathcal{D}$ is an infinite sequence

$$((\sigma_i, u_i, cons_i, Snd_i))_{i \in \mathbb{N}_0} \in \Sigma^\mathcal{D} \times \mathcal{D}(\mathcal{E}) \times 2^{\mathcal{D}(\mathcal{E})} \times 2^{(\mathcal{D}(\mathcal{E}) \cup \{*, +\}) \times \mathcal{D}(\mathcal{E})}$$


**The Language of a Model**

**Recall:** A UML model $\mathcal{M} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$ and a structure $\mathcal{D}$ denote a set $[\mathcal{M}]$ of (initial and consecutive) computations of the form

$$(\sigma_0, \varepsilon_0) \xrightarrow{a_0} (\sigma_1, \varepsilon_1) \xrightarrow{a_1} (\sigma_2, \varepsilon_2) \cdots$$

where

$$a_i = (\text{cons}_i, \text{Snd}_i, u_i) \in 2^{\mathcal{D}(\mathcal{E})} \times 2^{(\mathcal{D}(\mathcal{E}) \cup \{*, +\}) \times \mathcal{D}(\mathcal{E})} \times \mathcal{D}(\mathcal{E}).$$

For the connection between models and interactions, we **disregard** the configuration of the ether, and define as follows:

**Definition.** Let $\mathcal{M} = (\mathcal{C}, \mathcal{M}, \mathcal{E})$ be a UML model and $\mathcal{D}$ a structure. Then

$$\mathcal{L}(\mathcal{M}) := \{(\sigma_i, u_i, \text{cons}_i, \text{Snd}_i))_{i \in \mathbb{N}_0} \in (\Sigma^\mathcal{D} \times \tilde{A})^\omega \mid$$

$$\exists (\varepsilon_i)_{i \in \mathbb{N}_0} : (\sigma_0, \varepsilon_0) \xrightarrow{\text{cons}_0, \text{Snd}_0} (\sigma_1, \varepsilon_1) \cdots \in [\mathcal{M}]\}$$

is the **language** of $\mathcal{M}$. 
Signal and Attribute Expressions

- Let $\mathcal{S} = (\mathcal{T}, \mathcal{C}, V, \text{atr}, \mathcal{E})$ be a signature and $X$ a set of logical variables,

- The signal and attribute expressions $\text{Expr}_\mathcal{S}(\mathcal{E}, X)$ are defined by the grammar:

$$\psi ::= \text{true} \mid \text{expr} \mid E_{x,y}^1 \mid E_{x,y}^2 \mid \neg \psi \mid \psi_1 \lor \psi_2,$$

where $\text{expr} : \text{Bool} \in \text{Expr}_\mathcal{S}, E \in \mathcal{E}, x, y \in X$ (or keyword $\text{env}$, or $\ast$).

Satisfaction of Signal and Attribute Expressions

- Let $(\sigma, u, \text{cons, Snd}) \in \Sigma_{\mathcal{T}} \times \bar{\mathcal{A}}$ be a tuple consisting of system state, object identity, consume set, and send set.

- Let $\beta : X \rightarrow \mathcal{D}(\mathcal{E})$ be a valuation of the logical variables.

Then

- $(\sigma, u, \text{cons, Snd}) \models_\beta \text{true}$

- $(\sigma, u, \text{cons, Snd}) \models_\beta \text{expr}$ if and only if $I[\text{expr}](\sigma, \beta) = 1$

- $(\sigma, u, \text{cons, Snd}) \models_\beta \neg \psi$ if and only if not $(\sigma, \text{cons, Snd}) \models_\beta \psi$

- $(\sigma, u, \text{cons, Snd}) \models_\beta \psi_1 \lor \psi_2$ if and only if $(\sigma, u, \text{cons, Snd}) \models_\beta \psi_1$ or $(\sigma, u, \text{cons, Snd}) \models_\beta \psi_2$

- $(\sigma, u, \text{cons, Snd}) \models_\beta E_{x,y}^1$ if and only if $\beta(x) = u \land \exists e \in \text{dom}(\sigma) \cap \mathcal{D}(E) \cdot (e, \beta(y)) \in \text{Snd}$

- $(\sigma, u, \text{cons, Snd}) \models_\beta E_{x,y}^2$ if and only if $\beta(y) = u \land \text{cons} \cap \mathcal{D}(E) \neq \emptyset$
Satisfaction of Signal and Attribute Expressions

- Let \((σ, u, cons, Snd) \in Σ_{\mathcal{S}} \times \mathcal{A}\) be a tuple consisting of system state, object identity, consume set, and send set.
- Let \(β : X \rightarrow \mathcal{P}(\mathcal{E})\) be a valuation of the logical variables.

Then

- \((σ, u, cons, Snd) \models_β \text{true}\)
- \((σ, u, cons, Snd) \models_β \text{expr}\) if and only if \(I[expr](σ, β) = 1\)
- \((σ, u, cons, Snd) \models_β \neg ψ\) if and only if not \((σ, cons, Snd) \models_β ψ\)
- \((σ, u, cons, Snd) \models_β ψ_1 \lor ψ_2\) if and only if \((σ, u, cons, Snd) \models_β ψ_1\) or \((σ, u, cons, Snd) \models_β ψ_2\)
- \((σ, u, cons, Snd) \models_β E_x^t\) if and only if \(β(x) = u \land \exists e \in \text{dom}(σ) \cap \mathcal{P}(E) \bullet (e, β(y)) \in Snd\)
- \((σ, u, cons, Snd) \models_β E_y^t\) if and only if \(β(y) = u \land cons \cap \mathcal{P}(E) \neq \emptyset\)

Observation: semantics of models keeps track of sender and receiver at sending and consumption time, but we disregard the event identity (for simplicity).

Alternative: keep track of event identities between send and receive.

TBA over Signature

Definition. A TBA

\[
\mathcal{B} = (\text{Expr}_B(X), X, Q_{ini}, \rightarrow, Q_F)
\]

where \(\text{Expr}_B(X)\) is the set of signal and attribute expressions \(\text{Expr}_S(\mathcal{E}, X)\) over signature \(\mathcal{S}\) is called TBA over \(\mathcal{S}\).
Plan:

- Given an LSC $L$ with body
  
  $$(I, (\mathcal{I}, \leq), \sim, \mathcal{O}, \text{Msg}, \text{Cond}, \text{LocInv}),$$

- construct a TBA $B_L$, and

- define language $\mathcal{L}(L)$ of $L$ in terms of $\mathcal{L}(B_L)$,
  in particular taking activation condition and activation mode into account.

- Then $\mathcal{M} \models L$ (universal) if and only if $\mathcal{L}(\mathcal{M}) \subseteq \mathcal{L}(L)$.
  And $\mathcal{M} \models L$ (existential) if and only if $\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(L) \neq \emptyset$. 

$G = (N, E, f)$

$\sigma_0$, $\epsilon_0$

$
\begin{align*}
\pi &= (\sigma_i, \text{cons}_i, \text{Snd}_i)_{i \in \mathbb{N}} \\
\rightarrow \quad u_0 &= (\sigma_1, \epsilon_1) \cdots \quad w_{\pi}
\end{align*}
$

$\mathbf{UML}$

$\text{Model}$

$\mathcal{CD}, \mathcal{SM}$

$\mathcal{expr} = (\mathcal{F}, \mathcal{E}, \text{Var}, \mathcal{A})$

$\mathcal{SD}$

$\mathcal{SD}$

$\mathcal{SM} = (\Sigma, \alpha_{\mathcal{SD}}, \alpha_{\mathcal{SM}})$

$\mathcal{B} = (Q_{\mathcal{SD}}, q_0, \alpha_{\mathcal{SD}}, \alpha_{\mathcal{SM}}, F_{\mathcal{SD}})$

$\mathcal{B}_L$

$\mathcal{L}(\mathcal{L})$

$\mathcal{L}(B_L)$

$\mathcal{M} \models L$ (universal) if and only if $\mathcal{L}(\mathcal{M}) \subseteq \mathcal{L}(L)$.

And $\mathcal{M} \models L$ (existential) if and only if $\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(L) \neq \emptyset$. 

$\mathbf{Mathematics}$

$\mathbf{UML}$
Definition. Let \((I, (L, \leq), \sim, \triangleright, \text{Msg}, \text{Cond}, \text{LocInv})\) be an LSC body. A non-empty set \(\emptyset \neq C \subseteq L\) is called a cut of the LSC body iff

- it is downward closed, i.e. \(\forall l, l' \in C \land l \leq l' \implies l \in C\),
- it is closed under simultaneity, i.e. \(\forall l, l' \in C \land l \sim l' \implies l \in C\), and
- it comprises at least one location per instance line, i.e. \(\forall i \in I \exists l \in C \cdot i \circ l = i\).

A cut \(C\) is called hot, denoted by \(\theta(C) = \text{hot}\), if and only if at least one of its maximal elements is hot, i.e. if

\[\exists l \in C \cdot \theta(l) = \text{hot} \land \not\exists l' \in C \cdot l \prec l'\]

Otherwise, \(C\) is called cold, denoted by \(\theta(C) = \text{cold}\).

Cut Examples

\[\emptyset \neq C \subseteq L\] — downward closed — simultaneity closed — at least one loc. per instance line
References
