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:

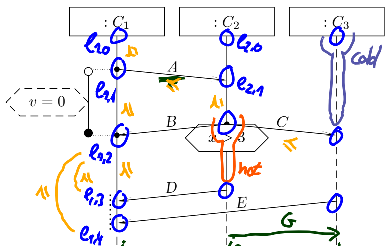
- **Educational Objectives:** Capabilities for following tasks/questions.
  - How is the semantics of LSCs constructed?
  - What is a cut, fired-set, etc.?
  - Construct the TBA for this LSC.
  - Give one example which (non-)trivially satisfies this LSC.
- **Content:**
  - Symbolic Automata
  - Firedset, Cut
  - Automaton construction
  - Transition annotations

Live Sequence Charts — Syntax

LSC Body: Abstract Syntax

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

$$(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv})$$



$I = \{ i_1, i_2, i_3 \}$

$\mathscr{L} = \{ \ell_{1,0}, \ell_{1,1}, \dots, \ell_{2,0}, \ell_{2,1}, \dots \}$

$\ell_{1,0} \preceq \ell_{1,1} \preceq \ell_{1,2} \dots$

$\ell_{1,1} \preceq \ell_{2,1}, \quad \dots$

- I is a finite set of **instance lines**,
- $(\mathscr{L}, \preceq)$ is a finite, non-empty, **partially ordered** set of **locations**; each $l \in \mathscr{L}$ is associated with a temperature $\theta(l) \in \Theta$ and an instance line $i_l \in I$,
- $\sim \subseteq \mathscr{L} \times \mathscr{L}$ is an **equivalence relation** on locations, the **simultaneity** relation,
- $\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr, \mathscr{E})$ is a signature,
- $\mathsf{Msg} \subseteq \mathscr{L} \times \mathscr{E} \times \mathscr{L}$ is a set of **asynchronous messages** with $(l, b, l') \in \mathsf{Msg}$ only if $l \preceq l'$, $\ell \nsim \ell'$
  ~~Not:~~ **instantaneous messages** — if $\ell \sim \ell'$
  could be mapped to method/operation calls.

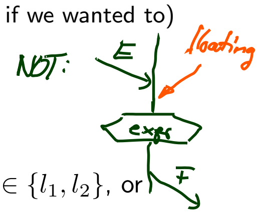
  $\mathsf{Msg} = \{ (\ell_{1,1}, A, \ell_{2,1}), \dots \}$
- $\mathsf{Cond} \subseteq (2^{\mathscr{L}} \setminus \emptyset) \times Expr_{\mathscr{S}} \times \Theta$ is a set of **conditions** where $Expr_{\mathscr{S}}$ are OCL expressions over $W = I \cup \{self\}$ with $(L, expr, \theta) \in \mathsf{Cond}$ only if $l \sim l'$ for all $l, l' \in L$,

  $\mathsf{Cond} = \{ (\{\ell_{2,2}\}, x > 3, hot), \dots \}$
- $\mathsf{LocInv} \subseteq \mathscr{L} \times \{\circ, \bullet\} \times Expr_{\mathscr{S}} \times \Theta \times \mathscr{L} \times \{\circ, \bullet\}$ is a set of **local invariants**,

  $\mathsf{LocInv} = \{ (\ell_{1,1}, \circ, v=0, cold, \ell_{2,2}, \bullet), \dots \}$

Well-Formedness

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



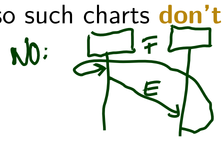
- For each location $l \in \mathscr{L}$, **if** l is the location of
  - a **condition**, i.e. $\exists\,(L, expr, \theta) \in$ Cond : $l \in L$, or
  - a **local invariant**, i.e. $\exists\,(l_1, i_1, expr, \theta, l_2, i_2) \in$ LocInv : $l \in \{l_1, l_2\}$, or

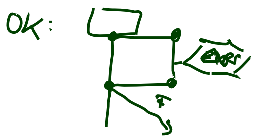
  then there is a location l' **equivalent** to l, i.e. $l \sim l'$, which is the location of

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

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

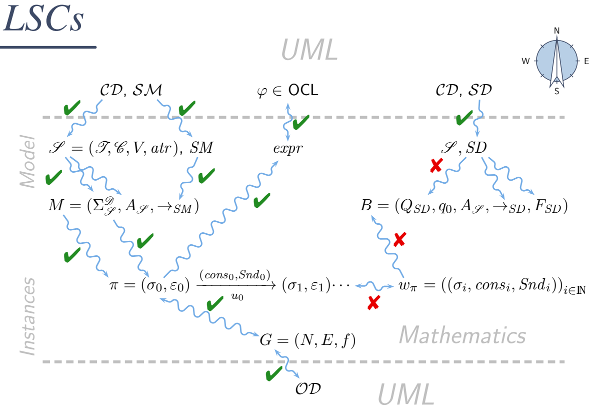
Note: if messages in a chart are **cyclic**, then there doesn't exist a partial order (so such charts **don't even have** an abstract syntax).





Live Sequence Charts — Semantics

TBA-based Semantics of LSCs



Plan:

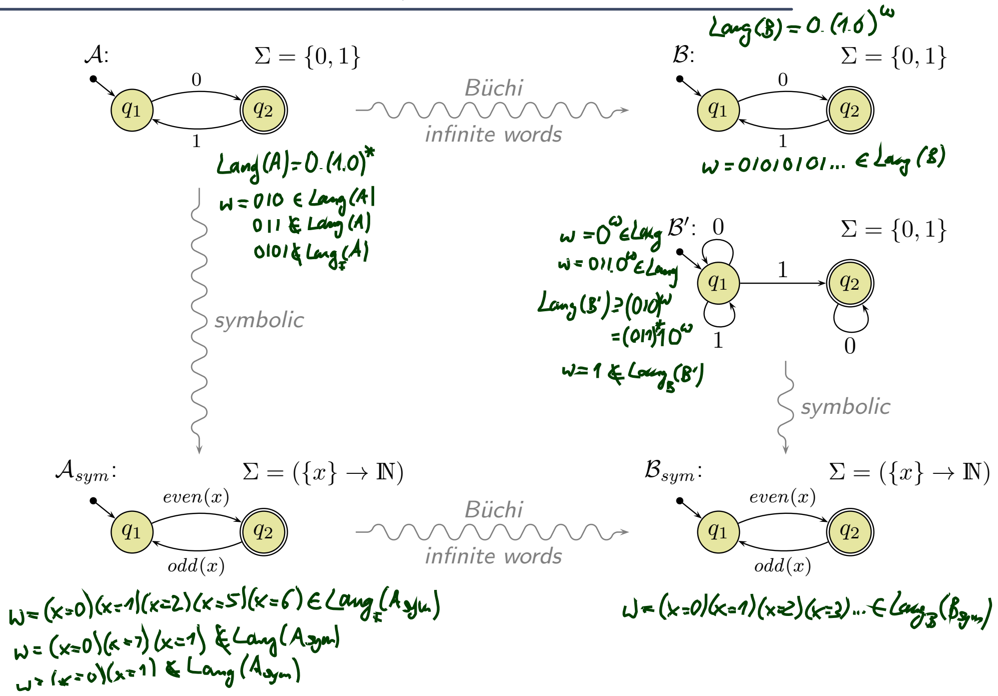
- Given an LSC L with body

$$(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}),$$

- construct a TBA $\mathcal{B}_L$, and
- define language $\mathcal{L}(L)$ of L **in terms of** $\mathcal{L}(\mathcal{B}_L)$,
  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 $\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(L) \neq \emptyset$.

Excursion: Büchi Automata

From Finite Automata to Symbolic Büchi Automata



Symbolic Büchi Automata

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

$$\mathcal{B} = (Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \rightarrow, Q_F)$$

where

- X is a set of logical variables,
- $Expr_{\mathcal{B}}(X)$ is a set of Boolean expressions over X,
- Q is a finite set of **states**,
- $q_{ini} \in Q$ is the initial state,
- $\rightarrow \subseteq Q \times Expr_{\mathcal{B}}(X) \times Q$ is the **transition relation**. Transitions (q, ψ, q') from q to q' are labelled with an expression $\psi \in Expr_{\mathcal{B}}(X)$.
- $Q_F \subseteq Q$ is the set of **fair** (or accepting) states.

Word

Definition. Let X be a set of logical variables and let $Expr_{\mathcal{B}}(X)$ be a set of Boolean expressions over X.

A set $(\Sigma, \cdot \models . \cdot)$ is called an **alphabet** for $Expr_{\mathcal{B}}(X)$ if and only if

- for each $\sigma \in \Sigma$,
- for each expression $expr \in Expr_{\mathcal{B}}$, and
- for each valuation $\beta : X \to \mathscr{D}(X)$ of logical variables to domain $\mathscr{D}(X)$,

either $\sigma \models_\beta expr$ **or** $\sigma \not\models_\beta expr$.

An **infinite sequence**

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

over $(\Sigma, \cdot \models . \cdot)$ is called **word** for $Expr_{\mathcal{B}}(X)$.

Run of TBA over Word

Definition. Let $\mathcal{B} = (Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \to, Q_F)$ be a TBA and

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

a word for $Expr_{\mathcal{B}}(X)$. An infinite sequence

$$\varrho = \underbrace{q_0, q_1}_{\psi_0}\underbrace{, q_2}_{\psi_1}, \ldots \in Q^\omega$$

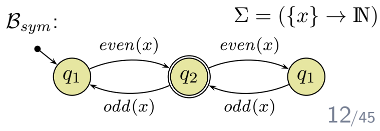
is called **run of $\mathcal{B}$ over w** under valuation $\beta : X \to \mathscr{D}(X)$ if and only if

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

$w = (x=0)(x=1)(x=2)\ldots$

$\varrho = q_1 q_2 q_1 \ldots$

Example:



The Language of a TBA

Definition.
We say TBA $\mathcal{B} = (Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \rightarrow, Q_F)$ **accepts** the word $w = (\sigma_i)_{i \in \mathbb{N}_0} \in (Expr_{\mathcal{B}} \rightarrow \mathbb{B})^\omega$ if and only if $\mathcal{B}$ **has** a run

$$\varrho = (q_i)_{i \in \mathbb{N}_0}$$

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

$$\forall\, i \in \mathbb{N}_0\ \exists\, j > i : q_j \in Q_F.$$

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

Language of UML Model

Words over Signature

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

$$(\sigma_i, u_i, cons_i, Snd_i)_{i \in \mathbb{N}_0} \in \Sigma_{\mathscr{S}}^{\mathscr{D}} \times \mathscr{D}(\mathscr{C}) \times 2^{\mathscr{D}(\mathscr{E})} \times 2^{(\mathscr{D}(\mathscr{E}) \,\dot\cup\, \{*,+\}) \times \mathscr{D}(\mathscr{C})}$$

$$(\sigma, \varepsilon) \xrightarrow[u]{(cons,\, Snd)} (\sigma', \varepsilon')$$

$$\rightsquigarrow$$

$$(\sigma, u, cons, Snd)$$

e.g. $(\sigma, u, \{e\}, \{(f, u)\})$

The Language of a Model

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

$$(\sigma_0, \varepsilon_0) \xrightarrow{a_0} (\sigma_1, \varepsilon_1) \xrightarrow{a_1} (\sigma_2, \varepsilon_2) \xrightarrow{a_2} \dots \text{ where}$$

$$a_i = (cons_i, Snd_i, u_i) \in \underbrace{2^{\mathscr{D}(\mathscr{E})} \times 2^{(\mathscr{D}(\mathscr{E}) \,\dot\cup\, \{*,+\}) \times \mathscr{D}(\mathscr{C})} \times \mathscr{D}(\mathscr{C})}_{=:\tilde{A}}.$$

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

Definition. Let $\mathcal{M} = (\mathscr{CD}, \mathscr{SM}, \mathscr{OD})$ be a UML model and $\mathscr{D}$ a structure. Then

$$\mathcal{L}(\mathcal{M}) := \{(\sigma_i, u_i, cons_i, Snd_i)_{i \in \mathbb{N}_0} \in (\Sigma_{\mathscr{S}}^{\mathscr{D}} \times \tilde{A})^\omega \mid$$
$$\exists\, (\varepsilon_i)_{i \in \mathbb{N}_0} : (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(cons_0, Snd_0)} (\sigma_1, \varepsilon_1) \cdots \in [\![\mathcal{M}]\!]\}$$

is the **language** of $\mathcal{M}$.

Signal and Attribute Expressions

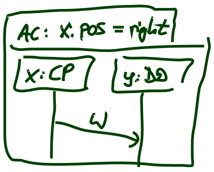
- Let $\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr, \mathscr{E})$ be a signature and X a set of logical variables,
- The signal and attribute expressions $Expr_{\mathscr{S}}(\mathscr{E}, X)$ are defined by the grammar:

$$\psi ::= \textit{true} \mid expr \mid E^{!}_{x,y} \mid E^{?}_{x,y} \mid \neg\psi \mid \psi_1 \vee \psi_2,$$

where $expr : Bool \in Expr_{\mathscr{S}}$, $E \in \mathscr{E}$, $x, y \in X$ (or keyword env, or $*$).

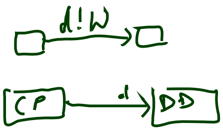
Satisfaction of Signal and Attribute Expressions

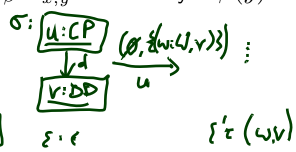
- Let $(\sigma, u, cons, Snd) \in \Sigma^{\mathscr{D}}_{\mathscr{S}} \times \tilde{A}$ be a tuple consisting of **system state**, **object identity**, **consume set**, and **send set**.
- Let $\beta : X \to \mathscr{D}(\mathscr{C})$ be a valuation of the logical variables.



Then

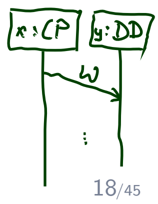
- $(\sigma, u, cons, Snd) \models_\beta \textit{true}$
- $(\sigma, u, cons, Snd) \models_\beta expr$ if and only if $I[\![expr]\!](\sigma, \beta) = 1$
- $(\sigma, u, cons, Snd) \models_\beta \neg\psi$ if and only if not $(\sigma, cons, Snd) \models_\beta \psi$
- $(\sigma, u, cons, Snd) \models_\beta \psi_1 \vee \psi_2$ if and only if
  $(\sigma, u, cons, Snd) \models_\beta \psi_1$ or $(\sigma, u, cons, Snd) \models_\beta \psi_2$
- $(\sigma, u, cons, Snd) \models_\beta E^{!}_{x,y}$ if and only if
  $\beta(x) = \underset{\mathscr{D}(\mathscr{C})}{u} \wedge \exists \underset{\mathscr{D}(E)}{e} \in \mathrm{dom}(\sigma) \cap \mathscr{D}(E) \bullet (e, \beta(y)) \in Snd$
- $(\sigma, u, cons, Snd) \models_\beta E^{?}_{x,y}$ if and only if $\beta(y) = u \wedge cons \cap \mathscr{D}(E) \neq \emptyset$





$(\sigma, u, \emptyset, \{(w, v)\}) \models_\beta W^{!}_{x,y}$

$\beta = \{ x \mapsto u, \; y \mapsto v \}$ $\in \mathscr{D}(CP)$, $\in \mathscr{D}(DD)$



Satisfaction of Signal and Attribute Expressions

- Let $(\sigma, u, cons, Snd) \in \Sigma_{\mathscr{S}}^{\mathscr{D}} \times \tilde{A}$ be a tuple consisting of **system state**, **object identity**, **consume set**, and **send set**.
- Let $\beta : X \to \mathscr{D}(\mathscr{C})$ be a valuation of the logical variables.

Then

- $(\sigma, u, cons, Snd) \models_\beta$ *true*
- $(\sigma, u, cons, Snd) \models_\beta expr$ if and only if $I[\![expr]\!](\sigma, \beta) = 1$
- $(\sigma, u, cons, Snd) \models_\beta \neg\psi$ if and only if not $(\sigma, cons, Snd) \models_\beta \psi$
- $(\sigma, u, cons, Snd) \models_\beta \psi_1 \vee \psi_2$ if and only if $(\sigma, u, cons, Snd) \models_\beta \psi_1$ or $(\sigma, u, cons, Snd) \models_\beta \psi_2$
- $(\sigma, u, cons, Snd) \models_\beta E^!_{x,y}$ if and only if $\beta(x) = u \wedge \exists\, e \in \mathrm{dom}(\sigma) \cap \mathscr{D}(E) \bullet (e, \beta(y)) \in Snd$
- $(\sigma, u, cons, Snd) \models_\beta E^?_{x,y}$ if and only if $\beta(y) = u \wedge cons \cap \mathscr{D}(E) \neq \emptyset$

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

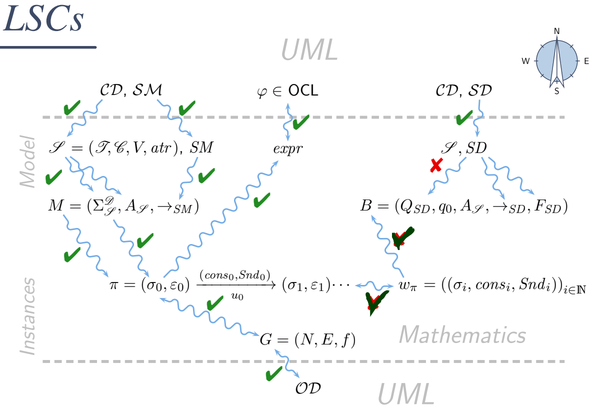
Definition. A TBA

$$\mathcal{B} = (Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \to, Q_F)$$

where $Expr_{\mathcal{B}}(X)$ is the set of **signal and attribute expressions** $Expr_{\mathscr{S}}(\mathscr{E}, X)$ over signature $\mathscr{S}$ is called **TBA over $\mathscr{S}$**.

Live Sequence Charts — Semantics

TBA-based Semantics of LSCs



Plan:

- Given an LSC L with body

$$(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}),$$

- construct a TBA $\mathcal{B}_L$, and
- define language $\mathcal{L}(L)$ of L **in terms of** $\mathcal{L}(\mathcal{B}_L)$,
  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 $\mathcal{L}(\mathcal{M}) \cap \mathcal{L}(L) \neq \emptyset$.

Formal LSC Semantics: It's in the Cuts!

Definition.
Let $(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv})$ be an LSC body.
A non-empty set $\emptyset \neq C \subseteq \mathscr{L}$ is called a **cut** of the LSC body iff

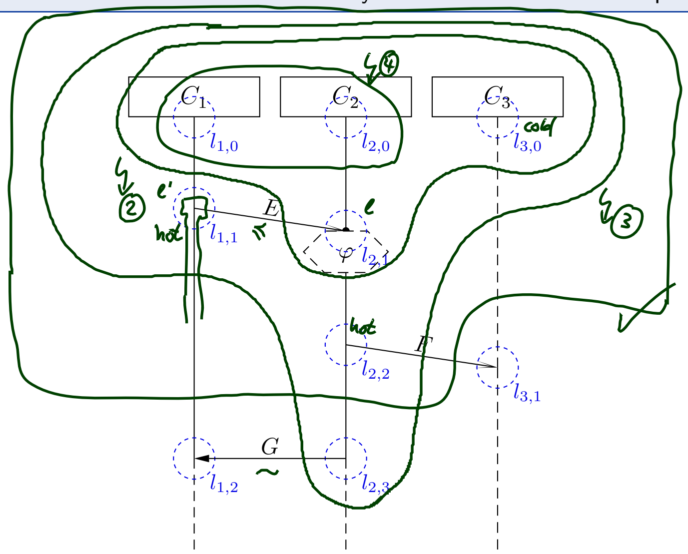
- it is **downward closed**, i.e. $\forall\, l, l' \bullet l' \in C \wedge l \preceq l' \implies l \in C$,
- it is **closed** under **simultaneity**, i.e.
$$\forall\, l, l' \bullet l' \in C \wedge l \sim l' \implies l \in C, \text{ and}$$
- it comprises at least **one location per instance line**, i.e.
$$\forall\, i \in I\ \exists\, l \in C \bullet i_l = i.$$

A cut C is called **hot**, denoted by $\theta(C) = \mathsf{hot}$, if and only if at least one of its maximal elements is hot, i.e. if
$$\exists\, l \in C \bullet \theta(l) = \mathsf{hot} \wedge \nexists\, l' \in C \bullet l \prec l'$$
Otherwise, C is called **cold**, denoted by $\theta(C) = \mathsf{cold}$.

Cut Examples

1 2 3 4

$\emptyset \neq C \subseteq \mathscr{L}$ — downward closed — simultaneity closed — at least one loc. per instance line



References

References

OMG (2011a). Unified modeling language: Infrastructure, version 2.4.1. Technical Report formal/2011-08-05.

OMG (2011b). Unified modeling language: Superstructure, version 2.4.1. Technical Report formal/2011-08-06.