Recall: Example

Recall: Live Sequence Charts Syntax

Recall: LSC Body – Abstract Syntax

Contents & Goals

Last Lecture:
- Reflection vs. construction description of behaviour
- Live Sequence Charts: syntax, intuition

This Lecture:
- Educational Objectives: Capabilities for following tasks/questions.
- What does this LSC mean?
- Are the UML models state machines consistent with the interaction?
- Please provide a UML model which is consistent with the LSC.
- What is: activation, hot/cold condition, pre-chart, etc.?

Content:
- Symbolic State Automata (TBA) and its (accepted) language
- LSC formal semantics.

Example

Let $I = \text{[not, cold]}$. An LSC body is a tuple

\[(I, \mathcal{C}, \mathcal{I}, \mathcal{E}, \mathcal{M}, \mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2)\]

where

- $I$ is a finite set of instance lines, usually named $\mathcal{I}$ in the case.
- $\mathcal{C}$ is a finite, non-empty, partially ordered set of locations, each of which is associated with a temperature $v \in \mathbb{T}$ and an instance line $i \in I$.
- $\mathcal{I} \subseteq \mathcal{C}$ is an equivalence relation on locations, the simultaneously relation,
- $\mathcal{E} = (\mathcal{V}, \mathcal{F})$ is a signature,
- $\mathcal{M} = (\mathcal{Z}, \mathcal{D})$ is a set of asynchronous messages with $(\mathcal{Z}, \mathcal{D}) \in \mathcal{M}$ only if $a \in \mathcal{I}$.

Not: instantaneous messages — could be linked to internal/operation calls

- Cond $C = (\mathcal{V}, \mathcal{F})$, which is a set of conditions with $(\; \mathcal{V}, \mathcal{F})$ in $\mathcal{M}$ only if $a \in \mathcal{I}$ for all $a$.

- Loclines $\subseteq \mathcal{C} \times \mathcal{E}$ represent where $\mathcal{E}$ is a set of local incident.
Recall: 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. $\exists (l, \text{expr}, \theta) \in \text{Cond} \mid l \in \mathcal{L}$,
  - a local invariant, i.e. $\exists (l, \text{inv}, \text{loc}) \in \text{LocInv} \mid l \in \mathcal{L}$,
- then there is a location $l'$ equivalent to $l$ which is the location of
  - a message, i.e. $\exists (l, \text{msg}, \text{dest}) \in \text{Msg} \mid l \in \mathcal{L}$,
  - an instance head, i.e. $l'$ is external wrt. $l$.

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

TBA-based Semantics of LSCs

Plan:
- Given an LSC $\mathcal{L}$ with body
  $$(E, \langle \mathcal{L} \rangle, \leftarrow \mathcal{S}, \text{Msg}, \text{Cond}, \text{LocInv})$$
- Construct a TBA $\mathcal{B}_\mathcal{L}$ — taking the cuts of $E$ as states.
- Define $\exists (l, \text{expr}, \theta) \in \text{Cond} \mid l \in \mathcal{L}$.
- in particular taking activation condition and activation mode into account.

Course Map

Live Sequence Charts Semantics

Formal LSC Semantics: It’s in the Cuts

- Let $(E, \langle \mathcal{L} \rangle, \leftarrow \mathcal{S}, \text{Msg}, \text{Cond}, \text{LocInv})$ be an LSC body.
- A non-empty set $B \subseteq \mathcal{L}$ is called a cut of the LSC body $\mathcal{L}$ if and only if
  - $B$ is downward closed, i.e. $\forall l' \in B \exists l \in B \mid l \leftarrow l'$,
  - $B$ is closed under simultaneity, i.e. $\forall l', l'' \in B \mid l' \leftarrow l'' \rightarrow l' \in B$ and
  - $B$ comprises at least one location per instance line, i.e. $\forall l \in \mathcal{L} \mid l \leftarrow l'$.

Formal LSC Semantics: It’s in the Cuts

- Let $(E, \langle \mathcal{L} \rangle, \leftarrow \mathcal{S}, \text{Msg}, \text{Cond}, \text{LocInv})$ be an LSC body.
- A non-empty set $B \subseteq \mathcal{L}$ is called a cut of the LSC body $\mathcal{L}$ if and only if
  - $B$ is downward closed, i.e. $\forall l' \in B \exists l \in B \mid l \leftarrow l'$,
  - $B$ is closed under simultaneity, i.e. $\forall l', l'' \in B \mid l' \leftarrow l'' \rightarrow l' \in B$ and
  - $B$ comprises at least one location per instance line, i.e. $\forall l \in \mathcal{L} \mid l \leftarrow l'$.
- A cut $C$ is called hot, denoted by $H(C) \equiv \text{hot}$, if and only if at least one of its non-empty elements is $\text{hot}$, i.e. if
  $$\exists l \in C : H(l)$$
  Otherwise, $C$ is called cold, denoted by $H(C) \equiv \text{cold}$.
A Successor Relation on Cats

The partial order of \((\mathcal{P}, \subseteq)\) and the simultaneity relation \(\sim\) induce a direct successor relation on cats \(\mathcal{C}\) as follows:

- Let \(\mathcal{C}, \mathcal{C}' \subseteq \mathcal{P}\) be cats. \(\mathcal{C}'\) is called direct successor of \(\mathcal{C}\) if \(\mathcal{C}' \subseteq \mathcal{C}\) via \(\mathcal{C}' \sim \mathcal{C}\).
- \(\mathcal{C} \neq \mathcal{C}'\).
- \(\mathcal{C}' \subseteq \mathcal{C}\) if for each message reception in \(\mathcal{C}'\), the corresponding sending is already in \(\mathcal{C}\).
- Locations in \(\mathcal{P}\) that lie on the same instance line, are pairwise ordered, i.e.,

  \[
  \forall \mathcal{F} \in \mathcal{P}. (\mathcal{F} \sim \mathcal{G} \Rightarrow \mathcal{F} \subseteq \mathcal{G})
  \]

- \(\mathcal{F}\) is immediately closed under simultaneity, \(\sim\).

In other words: locations in \(\mathcal{P}\) are direct \(\subseteq\) successors of locations in \(\mathcal{C}\), i.e.

\[
\forall \mathcal{F} \in \mathcal{P}. (\exists \mathcal{C} \subseteq \mathcal{P}. \mathcal{F} \subseteq \mathcal{C} \wedge \mathcal{F} \sim \mathcal{C})
\]

Excursus: Symbolic Büchi Automata (over Signature)

Definition: A Symbolic Büchi Automaton (SBA) is a tuple

\[\mathcal{SBA} = (\mathcal{Exp}_S, \mathcal{Q}, \mathcal{q}_0, \rightarrow_{\mathcal{SBA}})\]

where:

1. \(\mathcal{Exp}_S\) is a set of expressions over logical variables from \(\mathcal{X}\).
2. \(\mathcal{Q}\) is a finite set of states, \(\mathcal{q}_0\) the initial state.
3. \(\mathcal{Q} \subseteq \mathcal{Q}_S\) is the transition relation.
4. Transitions \(\mathcal{q} \rightarrow \mathcal{q}'\) from \(\mathcal{q}\) to \(\mathcal{q}'\) are labeled with a constraint \(\mathcal{v} \rightarrow \mathcal{v}'\).
5. \(\mathcal{Q} \subseteq \mathcal{Q}_S\) is the set of accepting states.
A word over $\mathcal{E}xpr_y$ is an infinite sequence of interpretations of $\mathcal{E}xpr_y$ such that

$$\left(\forall \sigma \in \Sigma \right) \left( \mathcal{E}xpr_y \sigma = 1 \right)$$

A run of $\mathcal{TBA}$ over a word $\omega$ is a sequence $(x_0, \ldots, x_i, \ldots) \in \Sigma^\omega$ such that

$$\left( \forall i \in \mathbb{N} \right) \left( x_i \right)$$

such that each state in a run is visited infinitely often.

We call the set $\mathcal{L}(\mathcal{TBA})$ of words over $\mathcal{TBA}$ that are accepted by $\mathcal{TBA}$ under $\mathcal{I}$ the language of $\mathcal{TBA}$. The language of $\mathcal{TBA}$ consists of the words $\omega$ over $\mathcal{TBA}$ that are accepted by $\mathcal{TBA}$ under $\mathcal{I}$.
Recall Idea: Accepting Words by Advancing the Cut

Let $w = (c_1, c_2, \ldots, c_n)$ be a word over $T'$ and $I$. Intuitively (and for non-distinguishing conditions), an LSC body $(I, T', \rightarrow, \rightarrow; M_0, G, G, [L])$ is supposed to accept $w$ (under valuation $I$) if and only if there exists a sequence $C_0 \rightarrow C_1 \rightarrow \cdots \rightarrow C_n$ and indices $i_1 < \cdots < i_n$ such that

- $C_0$ consists of the instance heads,
- for all $1 \leq j < n$, $c_{i_j} \in \text{cons}(C_{i_j})$ satisfies (under $I$) the hold condition of $C_{i_j}$,
- $C_n$ is cold,
- for all $i < n$, $(c_{i+1}, \ldots, c_n)$ satisfies (under $I$) the hold condition of $C_i$.

The Language of LSC Body

The language of the body

$L_w = (I, T', \rightarrow, \rightarrow; M_0, G, G, [L])$

of LSC $L$ is the language of the TBA $B_w = (\text{Expr}_{x,y}, I, \text{LocInv}, \text{Cond}, G, G, [L])$ with

- $\text{Expr}_{x,y} = \text{Expr}_{x,y}(T', I')$,
- $T'$ is the set of cutsof $[L, \rightarrow]$, $T'$ is the set of cold cuts,
- $I'$ consists of
  - loops $(q, q, \ldots)$
  - progress transitions $(q, q')$, and
  - legal exits $(q, x')$.

Signal and Integer Expressions

Let $T' = (T', \forall)$ be a signature and $X$ a set of logical variables. The signal and integer expressions $\text{Expr}_{x,y}(T', I')$ over $T'$ are defined by the grammar:

- $\phi ::= \top \mid \bot 
  \mid \text{loop} (q, q', \ldots) 
  \mid \text{trans} (q, q') 
  \mid \text{exit} (q, x')$

where $\text{loop} (q, q', \ldots)$, $\text{trans} (q, q')$, and $\text{exit} (q, x')$ are satisfied (under $I'$) the hold condition of $C_i$.

Saturation of Signal and Integer Expressions

Let $(c_1, c_2, \ldots, c_n) \in T'$ be a word over $T'$ and $I'$ and let $(X, T \rightarrow (Y \rightarrow W))$ be a valuation of the logical variables in $X$. The

- $(c_{i+1}, X, \ldots, c_n) \in T'$ if and only if $(c_{i+1}, X, \ldots, c_n) \in T'$

for

- $(\text{exit} (q, x'))$, $x' \in X$ and only if $(q, x') \in L$.
- $(\text{trans} (q, q'))$, $q, q' \in I'$ and only if $(q, q') \in L$.
- $(\text{loop} (q, q', \ldots))$, $q', q'' \in I'$ and only if $(q', q'') \in L$.
Satisfaction of Signal and Integer Expressions

Let $(s, \text{cons}, \text{Snd}) \in \{\text{Expr}, \text{Snd}, \text{Bv}\}$ be a letter of a word over $\mathcal{F}$ and $\mathcal{F}$ be a valuation of the

logical variables in $X$.

+ $(s, \text{cons}, \text{Snd}) \models s = \text{true}$
+ $(s, \text{cons}, \text{Snd}) \models s = \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s \neq s$
+ $(s, \text{cons}, \text{Snd}) \models s = \text{false}$ or $(s, \text{cons}, \text{Snd}) \models = \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s \neq = \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s \neq \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s = \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s \neq \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s = \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s \neq \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s = \text{false}$
+ $(s, \text{cons}, \text{Snd}) \models s \neq \text{false}$

Observation: If the semantics has "forgotten" the sender at some time, then we have to disregard it here (straightforwardly fixed if desired).

Example: TBA over Signal and Integer Expressions

Some More Helper Functions

- Constraints relevant when moving from $q$ to cut $q'$:
  \[ (i) \cup (\text{Expr}, \text{Snd}, \text{Bv}) \]

- Constraints relevant when moving from $q$ to cut $q'$:
  \[ (i) \cup (\text{Expr}, \text{Snd}, \text{Bv}) \]

- Constraints relevant when moving from $q$ to cut $q'$:
  \[ (i) \cup (\text{Expr}, \text{Snd}, \text{Bv}) \]

Even More Helper Functions

- Cold constraints relevant when moving from $q$ to cut $q'$:
  \[ \text{cold}(q, q') = \{ q \mid \exists \sigma \neq q', \sigma \not\in \mathcal{F}, \sigma \in \mathcal{F} \} \]
  \[ \text{LocInv} \cup (\text{Expr}, \text{Snd}, \text{Bv}) \]

Some Helper Functions

- Messages of location:
  \[ \mathcal{L}(i) = \{ \mathcal{L}(i), l, l \} \]

Recall: Intuition

- \[ \mathcal{L}(i) = \{ \mathcal{L}(i), l, l \} \] with
  - \( \mathcal{L}(i) = \) the set of cuts of $\mathcal{F}$, $\mathcal{F}$ is the instance heads cut,
  - $\mathcal{F} = \{ q \mid \exists \sigma \not\in \mathcal{F}, \sigma \in \mathcal{F} \}$ is the set of cold cuts,
  - $\mathcal{F} = \{ q \mid \exists \sigma \not\in \mathcal{F}, \sigma \in \mathcal{F} \}$ is the set of cold cuts,
  - $\mathcal{F} = \{ q \mid \exists \sigma \not\in \mathcal{F}, \sigma \in \mathcal{F} \}$ is the set of cold cuts,
  - $\mathcal{F} = \{ q \mid \exists \sigma \not\in \mathcal{F}, \sigma \in \mathcal{F} \}$ is the set of cold cuts,
  - $\mathcal{F} = \{ q \mid \exists \sigma \not\in \mathcal{F}, \sigma \in \mathcal{F} \}$ is the set of cold cuts,
Interactions as Reflective Description

- In UML, reflective (temporal) descriptions are subsumed by interactions.
- A UML model \( M = (\mathcal{P}, \mathcal{A}, \mathcal{C}, \mathcal{F}) \) has a set of interactions \( \mathcal{F} \).
- An interaction \( I \in \mathcal{F} \) can be (OMG claim: equivalently) diagrammed as
  - sequence diagram, timing diagram, or
  - communication diagram (formerly known as collaboration diagram).
**Why Sequence Diagrams?**

*Most Prominent: Sequence Diagrams — with long history:
  - Message Sequence Charts, standardized by the ITU in different versions, often accused to lack a formal semantics.
  - Sequence Diagrams of UML 1.x

Most severe drawbacks of these formalisms:
  - unclear interpretation: example scenario or invariant?
  - unclear activation: what triggers the requirement?
  - unclear progress requirement: must all messages be observed?
  - conditions merely comments
  - no means to express forbidden scenarios

---

**Thus: Live Sequence Charts**

- SDs of UML 2.x address some issues, yet the standard exhibits inconsistencies and even contradictions [Harel and Maoz, 2007, Störrle, 2003]
- For the lecture, we consider Live Sequence Charts (LSCs) [Damm and Harel, 2003, Klöse, 2003, Klöse and Marelly, 2003], who have a common fragment with UML 2.x SDs [Harel and Maoz, 2007]
- Modelling guideline: stick to that fragment.

---

**Side Note: Protocol Statemachines**

Same direction: call orders on operations
  - “For each C instance, method f() shall only be called after g() but before h()”

Can be formalized with protocol state machines.

---

**References**


