Contents & Goals

Last Lecture:
- Symbolic Büchi Automata
- Language of a UML Model
- Cuts

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:
  - Cut Examples, Firedset
  - Automaton construction
  - Transition annotations
  - Forbidden scenarios
Plan:

- Given an LSC \( L \) with body

\[
(I, (L, \leq), \sim, \mathcal{S}, \text{Msg}, \text{Cond}, \text{LocInv}),
\]

- 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 \( \mathcal{L}(\mathcal{M}) \cap \mathcal{L}(L) \neq \emptyset \).
**Formal LSC Semantics: It’s in the Cuts!**

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

- it is **downward closed**, i.e. \( \forall l, l' \bullet l' \in C \land l \preceq l' \implies l \in C \),
- it is **closed** under **simultaneity**, i.e.
  \[
  \forall l, l' \bullet l' \in C \land l \sim l' \implies l \in C , \text{ and}
  \]
- 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 \( \theta(C) = \text{hot} \), if and only if at least one of its maximal elements is hot, i.e. if

\[
\exists l \in C \bullet \theta(l) = \text{hot} \land \nexists l' \in C \bullet l \prec l'.
\]

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

**Cut Examples**

\[
\emptyset \neq C \subseteq \mathcal{L} \quad \text{— downward closed} \quad \text{— simultaneity closed} \quad \text{— at least one loc. per instance line}
\]
The partial order of \((\mathcal{L}, \preceq)\) and the simultaneity relation \(\sim\) induce a **direct successor relation** on cuts of \(\mathcal{L}\) as follows:

**Definition.** Let \(C, C' \subseteq \mathcal{L}\) be cuts of an LSC body with locations \((\mathcal{L}, \preceq)\) and messages \(\text{Msg}\).

\(C'\) is called **direct successor** of \(C\) via **fired-set** \(F\), denoted by \(C \leadsto_F C'\), if and only if

- \(F \neq \emptyset\),
- \(C' \setminus C = F\),
- for each asynchronous (!) message reception in \(F\), the corresponding sending is already in \(C\),
- locations in \(F\), that lie on the same instance line, are pairwise unordered, i.e.

\[
\forall l, l' \in \text{Msg}, l \neq l' : l' \in F \implies l \in C, \text{ and}
\]

\[
\forall l, l' \in F : l \neq l' \land i_l = i_{l'} \implies l \not\preceq l' \land l' \not\preceq l
\]
Properties of the Fired-set

C \rightsideset{\sim F}{\leftrightarrow} C' if and only if

- F \neq \emptyset,
- C' \setminus C = F,
- \forall (l, E, l') \in \text{Msg}, l \neq l' \land l' \in F \implies l \in C, and
- \forall l, l' \in F : l \neq l' \land i_l = i_{l'} \implies l \not\preceq l' \land l' \not\preceq l

- **Note:** F is closed under simultaneity.
- **Note:** locations in F are direct \preceq-successors of locations in C, i.e.

\[ \forall l' \in F \exists l \in C : l \prec l' \land \exists l'' \in C : l'' \prec l' \quad (\forall) \]

Successor Cut Example

C \cap F = \emptyset \implies \text{C is a cut — only direct \preceq-successors — same instance line on front pairwise unordered — sending of asynchronous reception already in}
Successor Cut Example

\[ C \cap F = \emptyset \quad \text{only direct \( \prec \)-successors} \quad \text{same instance line on front} \quad \text{pairwise unordered} \quad \text{sending of asynchronous reception already in} \]

Language of LSC Body: Example

The TBA \( B_L \) of LSC \( L \) over \( \Phi \) and \( \mathcal{E} \) is \( (\text{Expr}_\Phi(X), X, Q, q_{init}, \rightarrow, Q_F) \) with

- \( Q \) is the set of cuts of \( L \), \( q_{init} \) is the instance heads cut,
- \( \text{Expr}_\Phi(X) = \text{Expr}_\Phi(\mathcal{E}, X) \) (for considered signature \( \mathcal{E} \)),
- \( \rightarrow \) consists of loops, progress transitions (by \( \rightarrow_F \)), 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.
Recall: The TBA $B(L)$ of LSC $L$ is $(\text{Expr}_B(X), X, Q, q_{\text{ini}}, \rightarrow, Q_F)$ with

- $Q$ is the set of cuts of $L$, $q_{\text{ini}}$ is the instance heads cut,
- $\text{Expr}_B(X) = \text{Expr}_\mathcal{S}(\delta, X)$ (for considered signature $\mathcal{S}$),
- $\rightarrow \subseteq Q \times \text{Expr}_\mathcal{S}(\delta, X) \times Q$ consists of
  - loops, progress transitions (by $\rightarrow_F$), and legal exits (cold conditions / cold local invariants),
- $F = \{ C \in Q \mid \Theta(C) = \text{cold} \lor C = L \}$ is the set of cold cuts.

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

$$
\rightarrow = \{ (q, \text{expr}_{\text{loop}}(q), q) \mid q \in Q \} \cup \{ (q, \text{expr}_{\text{prog}}(q, q'), q') \mid q \rightarrow_F q' \} \cup \{ (q, \text{expr}_{\text{exit}}(q), L) \mid q \in Q \}
$$
**Loop Condition**

\[ \text{expr}_{\text{loop}}(q) = \text{expr}_{\text{Msg}}(q) \wedge \text{expr}_{\text{LocInv}}^{\text{hot}}(q) \wedge \text{expr}_{\text{Cond}}^{\text{hot}}(q) \]

- \( \text{expr}_{\text{Msg}}(q) = \neg \bigvee_{1 \leq i \leq n} \text{expr}_{\text{Msg}}(q, q_i) \wedge \bigwedge_{\text{strict} \implies \text{expr} \in \text{E}!? \cap \text{Msg}(L)} \neg \text{expr} \)
- \( \text{expr}_{\text{LocInv}}^{\text{hot}}(q) = \bigwedge_{l=(l', \phi, l', \phi') \in \text{LocInv}, \Theta(l)=\theta} \text{expr}_{\text{LocInv}}^{\theta}(q) = \bigwedge_{\lambda=(\ell, \phi, \ell, \phi) \in \text{LocInv}, \Theta(\lambda)=\theta} \text{expr}_{\text{LocInv}}^{\theta}(q) \)

A location \( l \) is called **front location** of cut \( C \) if and only if there is no location \( l' \) such that \( l \prec l' \).

Local invariant \((l_0, t_0, \phi, l_1, t_1)\) is **active** at cut \((1)\) \( q \) if and only if \( l_0 \leq l \leq l_1 \) for some front location \( l \) of cut \((1)\) \( q \).

**Progress Condition**

\[ \text{expr}_{\text{Cond}}^{\text{hot}}(q, q_i) = \text{expr}_{\text{Msg}}^{\text{hot}}(q, q_n) \wedge \text{expr}_{\text{LocInv}}^{\text{hot}}(q) \wedge \text{expr}_{\text{LocInv}}^{\text{hot}}(q_0) \]

- \( \text{expr}_{\text{Msg}}^{\text{hot}}(q, q_i) = \bigwedge_{\text{expr} \in \text{Mag}(q_i \setminus q)} \text{expr} \wedge \bigwedge_{j \neq i} \text{expr} \in (\text{Mag}(q_j \setminus q) \setminus \text{Mag}(q_i \setminus q)) \neg \text{expr} \)
- \( \text{expr}_{\text{LocInv}}^{\text{hot}}(q, q_i) = \bigwedge_{\gamma=(L, \phi) \in \text{Cond}, \Theta(\gamma)=\theta} \text{expr} \in (\text{E}!? \cap \text{Msg}(L)) \neg \text{expr} \)

Local invariant \((l_0, t_0, \phi, l_1, t_1)\) is **active** at \( q \) if and only if

- \( l_0 < l < l_1 \), or
- \( l = l_0 \wedge t_0 = \cdot \), or
- \( l = l_1 \wedge t_1 = \cdot \)

for some front location \( l \) of cut \((1)\) \( q \).
Example

Finally: The LSC Semantics

A full LSC $L = ((I, (\mathcal{L}, \preceq), \leadsto, \mathcal{S}, \text{Msg}, \text{Cond}, \text{LocInv}), ac_0, am, \Theta_L)$ consist of

- **body** $(I, (\mathcal{L}, \preceq), \leadsto, \mathcal{S}, \text{Msg}, \text{Cond}, \text{LocInv})$,
- **activation condition** $ac_0 : \text{Bool} \in \text{Expr}$, **strictness flag** strict (otherwise called permissive)
- **activation mode** $am \in \{\text{initial}, \text{invariant}\}$,
- **chart mode** existential ($\Theta_L = \text{cold}$) or **universal** ($\Theta_L = \text{hot}$).

Concrete syntax:
Finally: The LSC Semantics

A full LSC $L = ((I, \mathcal{L}, \preceq), \sim, \mathcal{A}, \text{Msg}, \text{Cond}, \text{LocInv}), ac_0, am, \Theta)$ consist of

- **body** $(I, (\mathcal{L}, \preceq), \sim, \mathcal{A}, \text{Msg}, \text{Cond}, \text{LocInv})$,
- **activation condition** $ac_0 : \text{Bool} \in \text{Expr}_\mathcal{A}$, **strictness flag** $\text{strict}$ (otherwise called permisive)
- **activation mode** $am \in \{\text{initial}, \text{invariant}\}$,
- **chart mode** existential ($\Theta_L = \text{cold}$) or universal ($\Theta_L = \text{hot}$).

A set of words $W \subseteq (\Sigma_D \times \bar{\mathcal{A}})^\omega$ is accepted by $L$ if and only if

<table>
<thead>
<tr>
<th>$\Theta_L$</th>
<th>$am = \text{initial}$</th>
<th>$am = \text{invariant}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cold</td>
<td>$\exists w \in W \exists \beta \bullet w^h \models_{\beta} ac \land w^h \models_{\beta} \text{expr}^{\text{Cond}}(\emptyset, C_0) \land w/1 \in \mathcal{L}(B(L))$</td>
<td>$\exists w \in W \exists \beta \exists k \in \mathbb{N}<em>0 \bullet w^k \models</em>{\beta} ac \land w^k \models_{\beta} \text{expr}^{\text{Cond}}(\emptyset, C_0) \land w/k+1 \in \mathcal{L}(B(L))$</td>
</tr>
<tr>
<td>hot</td>
<td>$\forall w \in W \forall \beta \bullet w^h \models_{\beta} ac \implies$$w^h \models_{\beta} \text{expr}^{\text{Cond}}(\emptyset, C_0) \land w/1 \in \mathcal{L}(B(L))$</td>
<td>$\forall w \in W \forall \beta \forall k \in \mathbb{N}<em>0 \bullet w^k \models</em>{\beta} ac \implies$$w^k \models_{\beta} \text{expr}^{\text{Cond}}(\emptyset, C_0) \land w/k+1 \in \mathcal{L}(B(L))$</td>
</tr>
</tbody>
</table>

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

## Activation Condition

![Activation Condition Diagram](image-url)
Existential LSC Example: Buy A Softdrink

LSC: buy softdrink
AC: true
AM: invariant I: permissive

User \rightarrow Vend. Ma.

E1 \rightarrow SOFT
\downarrow SOFT

Existential LSC Example: Get Change

LSC: get change
AC: true
AM: invariant I: permissive

User \rightarrow Vend. Ma.

E1 \rightarrow SOFT
\downarrow SOFT
\downarrow chg-C50
A full LSC \( L = (PC, MC, ac_0, am, \Theta_L) \) actually consist of

- pre-chart \( PC = (I_P, (\mathcal{L}_P, \preceq_P), \sim_P, \mathcal{T}, \text{Msg}_P, \text{Cond}_P, \text{LocInv}_P) \) (possibly empty),

- main-chart \( MC = (I_M, (\mathcal{L}_M, \preceq_M), \sim_M, \mathcal{T}, \text{Msg}_M, \text{Cond}_M, \text{LocInv}_M) \) (non-empty),

- activation condition \( ac_0 : \text{Bool} \in \text{Expr}_T \), strictness flag \( \text{strict} \) (otherwise called permissive)

- activation mode \( am \in \{\text{initial}, \text{invariant}\}, \)

- chart mode existential \((\Theta_L = \text{cold})\) or universal \((\Theta_L = \text{hot})\).
### Pre-Charts Semantics

#### Universal LSC: Example

<table>
<thead>
<tr>
<th>Condition Type</th>
<th>Initial</th>
<th>Invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cold</strong></td>
<td>$\exists w \in W \exists \beta \exists m \in \mathbb{N}_0 \cdot w^0 \models \beta \text{ ac}$</td>
<td>$\exists w \in W \exists \beta \exists m &lt; k \in \mathbb{N} \cdot w^k \models \beta \text{ ac}$</td>
</tr>
<tr>
<td></td>
<td>$\land w^0 \models \text{expr}_{\text{cold}}(\emptyset, C^0_M)$</td>
<td>$\land w^k \models \text{expr}_{\text{cold}}(\emptyset, C^k_N)$</td>
</tr>
<tr>
<td></td>
<td>$\land w^1, \ldots, w^m \in \mathcal{L}(B(PC))$</td>
<td>$\land w^k + 1, \ldots, w^m \in \mathcal{L}(B(PC))$</td>
</tr>
<tr>
<td></td>
<td>$\land w^m + 1 \models \text{expr}_{\text{cold}}(\emptyset, C^m_M)$</td>
<td>$\land w^m + 1 \models \text{expr}_{\text{cold}}(\emptyset, C^m_N)$</td>
</tr>
<tr>
<td></td>
<td>$\land w^m + 1 \in \mathcal{L}(B(MC))$</td>
<td>$\land w^m + 1 \in \mathcal{L}(B(MC))$</td>
</tr>
</tbody>
</table>

#### Diagrams

- **Pre-chart**
- **Main chart**
Universal LSC: Example

Forbidden Scenario Example: Don’t Give Two Drinks
Note: Scenarios and Acceptance Test

- **Existential LSCs** may hint at test-cases for the acceptance test!
  (+: as well as (positive) scenarios in general, like use-cases)

- **Universal LSCs** (and negative/anti-scenarios) in general need exhaustive analysis!
  (Because they require that the software never ever exhibits the unwanted behaviour.)

References
References
