Software Design, Modelling and Analysis in UML

Lecture 20: Live Sequence Charts IV

2017-02-02

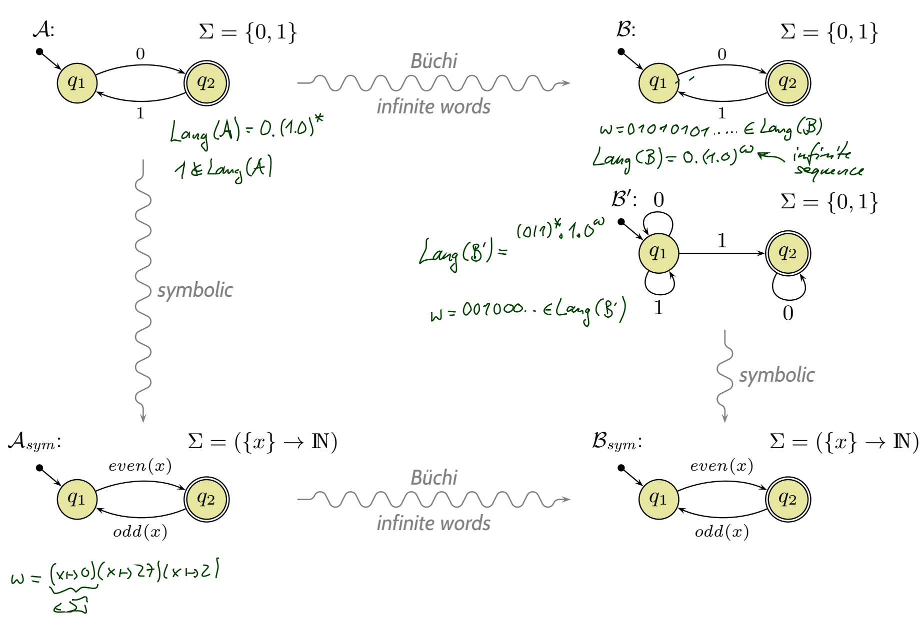
Prof. Dr. Andreas Podelski, **Dr. Bernd Westphal**

Albert-Ludwigs-Universität Freiburg, Germany

Content

Excursion: Büchi Automata

From Finite Automata to Symbolic Büchi Automata



Symbolic Büchi Automata

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

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

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,
- $\to \subseteq Q \times Expr_{\mathcal{B}}(X) \times Q$ 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,

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

(σ **satisfies** (or does not satisfy) $expr$ under valuation β)

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

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

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

$$\varrho = q_0, q_1, q_2, \dots \in Q^\omega$$

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

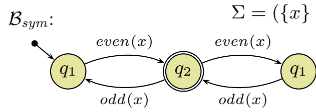
- $q_0 = q_{ini}$,
- for each $i \in \mathbb{N}_0$ there is a transition

$$(q_i, \psi_i, q_{i+1}) \in \rightarrow$$

  such that $\sigma_i \models_\beta \psi_i$.

$\Sigma = (\{x\} \rightarrow \mathbb{N})$

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

$$w = (\sigma_i)_{i \in \mathbb{N}_0} \in (Expr_{\mathcal{B}} \rightarrow \mathbb{B})^\omega$$

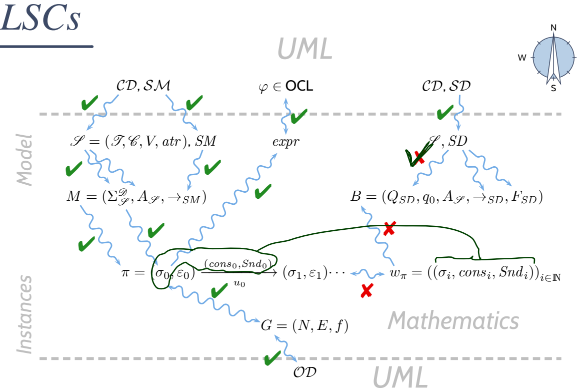
if and only if $\mathcal{B}$ **has** a run

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

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** $\mathcal{B}$.



Plan:

(i) Given an LSC $\mathscr{L}$ with body

$$((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta),$$

(ii) construct a TBA $\mathcal{B}_\mathscr{L}$, and

(iii) define language $\mathcal{L}(\mathscr{L})$ of $\mathscr{L}$ **in terms of** $\mathcal{L}(\mathcal{B}_\mathscr{L})$,
in particular taking activation condition and activation mode into account.

(iv) define language $\mathcal{L}(\mathcal{M})$ of a UML model.

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

Language of UML Model

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

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

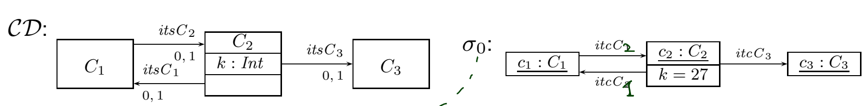
$$a_i = (cons_i, Snd_i, u_i) \in \underbrace{2^{\mathscr{D}(\mathscr{E})} \times 2^{(\mathscr{D}(\mathscr{E}) \,\dot\cup\, \{*,+\}) \times \mathscr{D}(\mathscr{C})} \times \mathscr{D}(\mathscr{C})}_{=:\tilde{A}}.$$

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 $\mathcal{M}$.

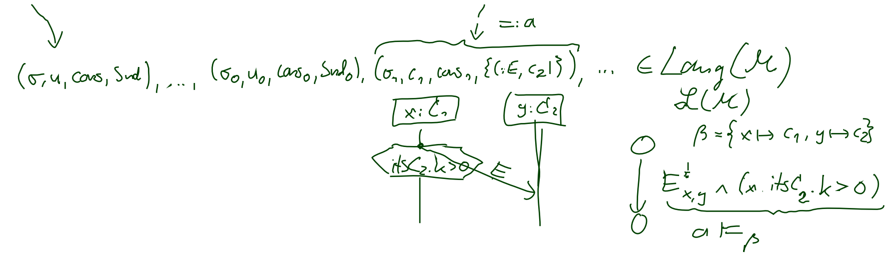
Example: Language of a Model

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



$(\sigma, \varepsilon) \xrightarrow[u]{(cons, Snd)} \dots \to (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(cons_0, Snd_0)} (\sigma_1, \varepsilon_1) \xrightarrow[c_1]{(cons_1, \{(:E, c_2)\})} (\sigma_2, \varepsilon_2) \xrightarrow[c_2]{(\{:E\}, Snd_2)}$

$(\sigma_3, \varepsilon_3) \xrightarrow[c_2]{(cons_3, \{(:F, c_3)\})} (\sigma_4, \varepsilon_4) \xrightarrow[c_2]{(cons_4, \{(G(), c_1)\})} (\sigma_5, \varepsilon_5) \xrightarrow[c_3]{(\{:F\}, Snd_5)} (\sigma_6, \varepsilon_6) \to \cdots$



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

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

- The language $\mathcal{L}(\mathcal{M})$ of a UML model $\mathcal{M} = (\mathscr{CD}, \mathscr{SM}, \mathscr{OD})$ is a word over the signature $\mathscr{S}(\mathscr{CD})$ induced by $\mathscr{CD}$ and $\mathscr{D}$, given structure $\mathscr{D}$.

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.

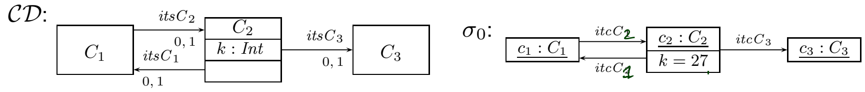
Then

- $(\sigma, u, cons, Snd) \models_\beta$ *true*
- $(\sigma, u, cons, Snd) \models_\beta \psi$ if and only if $I[\![\psi]\!](\sigma, \beta) = 1$
- $(\sigma, u, cons, Snd) \models_\beta \neg\psi$ 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 $\beta(x) = u \wedge \exists\, e \in \mathscr{D}(E) \bullet (e, \beta(y)) \in Snd$
- $(\sigma, u, cons, Snd) \models_\beta E^?_{x,y}$ if and only if $\beta(y) = u \wedge cons \subset \mathscr{D}(E)$

Observation: we don't use all information from the computation path.

We could, e.g., also keep track of event identities between send and receive.

Example: Model Language and Signal / Attribute Expresions



$$(\sigma, \varepsilon) \xrightarrow[u]{(cons, Snd)} \cdots \to (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(cons_0, Snd_0)} (\sigma_1, \varepsilon_1) \xrightarrow[c_1]{(cons_1, \{(:E, c_2)\})} (\sigma_2, \varepsilon_2) \xrightarrow[c_2]{(\{:E\}, Snd_2)}$$

$$(\sigma_3, \varepsilon_3) \xrightarrow[c_2]{(cons_3, \{(:F, c_3)\})} (\sigma_4, \varepsilon_4) \xrightarrow[c_2]{(cons_4, \{(G(), c_1)\})} (\sigma_5, \varepsilon_5) \xrightarrow[c_3]{(\{:F\}, Snd_5)} (\sigma_6, \varepsilon_6) \to \cdots$$

- $\beta = \{x \mapsto c_1, y \mapsto c_2, z \mapsto c_3\}$
- $(\sigma_0, u_0, cons_0, Snd_0) \models_\beta y.k > 0$ ✓
- $(\sigma_0, u_0, cons_0, Snd_0) \models_\beta x.k > 0$ (NOT WELL-TYPED)
- $(\sigma_1, c_1, cons_1, \{(: E, c_2)\}) \models_\beta E^!_{x,y}$ ✓ ($c_1 = \beta(x)$, $c_2 = \beta(y)$)
- $(\sigma_1, c_1, cons_1, \{(: E, c_2)\}) \models_\beta F^!_{x,y}$ ✗ (F is not E)
- $\cdots \models_\beta E^?_{x,y}$ ✓
- We set $(\sigma_4, c_2, cons_4, \{G(), c_1\}) \models_\beta G^!_{y,x} \wedge G^?_{y,x}$ (triggered operation or method call).

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 $\mathscr{S}$**.

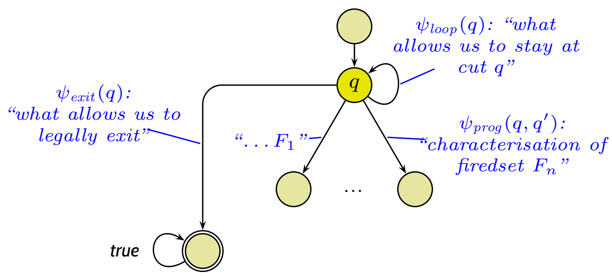
TBA Construction Principle

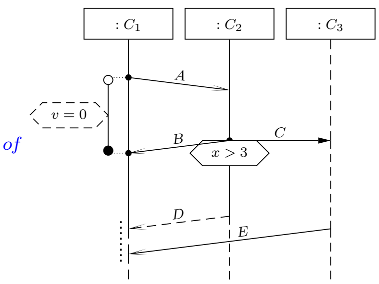
Recall: The TBA $\mathcal{B}(\mathscr{L})$ of LSC $\mathscr{L}$ is $(Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \to, Q_F)$ with

- Q is **the set of cuts** of $\mathscr{L}$, q_{ini} is the **instance heads** cut,
- $Expr_{\mathcal{B}} = \Phi \cup \mathscr{E}_{!?}(X)$, signal/attribute expressions
- $\to$ consists of **loops**, **progress transitions** (from $\rightsquigarrow_F$), and **legal exits** (cold cond./local inv.),
- $F = \{C \in Q \mid \Theta(C) = \text{cold} \vee C = L\}$ is the set of cold cuts.

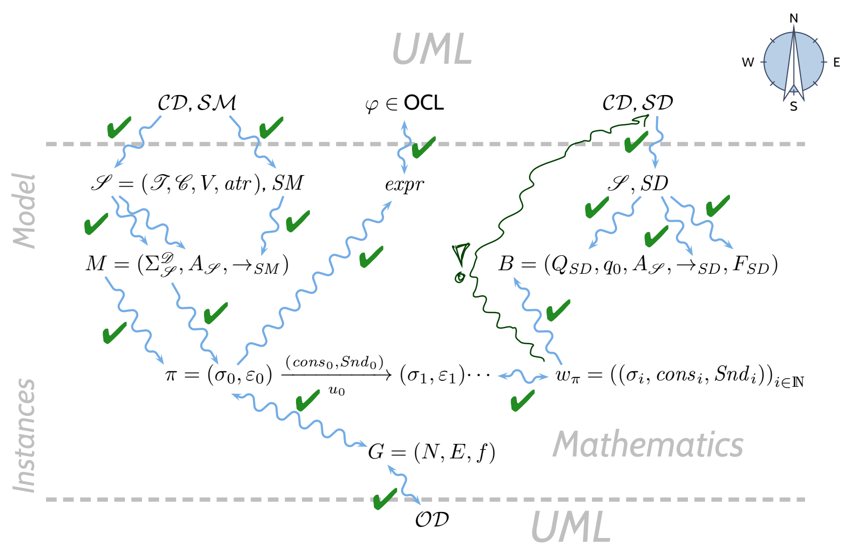
So in the following, we "only" need to construct the transitions' labels:

$$\to = \{(q, \psi_{loop}(q), q) \mid q \in Q\} \cup \{(q, \psi_{prog}(q, q'), q') \mid q \rightsquigarrow_F q'\} \cup \{(q, \psi_{exit}(q), L) \mid q \in Q\}$$





Course Map



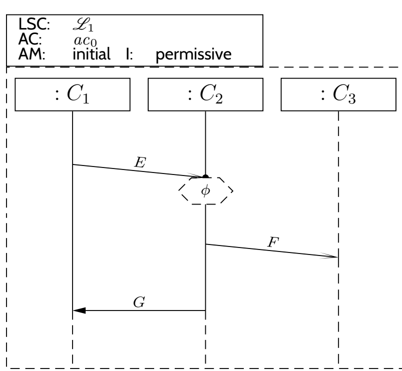
Live Sequence Charts — Full LSC Semantics

Full LSCs

A **full LSC** $\mathscr{L} = (((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta), ac_0, am, \Theta_{\mathscr{L}})$ consists of

- **body** $((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta)$,
- **activation condition** $ac_0 \in Expr_{\mathscr{L}}$,
- **strictness flag** $strict$ (if *false*, $\mathscr{L}$ is called **permissive**)
- **activation mode** *am* $\in \{\text{initial}, \text{invariant}\}$,
- **chart mode** **existential** ($\Theta_{\mathscr{L}} = \text{cold}$) or **universal** ($\Theta_{\mathscr{L}} = \text{hot}$).

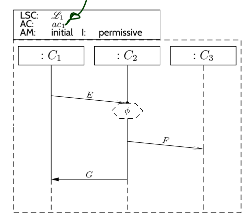
Concrete syntax:



Full LSCs

A **full LSC** $\mathscr{L} = (((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta), ac_0, am, \Theta_{\mathscr{L}})$ consists of

- **body** $((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta)$,
- **activation condition** $ac_0 \in Expr_{\mathscr{L}}$,
- **strictness flag** $strict$ (if *false*, $\mathscr{L}$ is called **permissive**)
- **activation mode** *am* $\in$ {initial, invariant},
- **chart mode existential** ($\Theta_{\mathscr{L}}$ = cold) or **universal** ($\Theta_{\mathscr{L}}$ = hot).

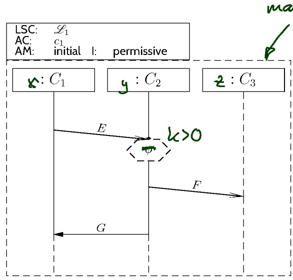


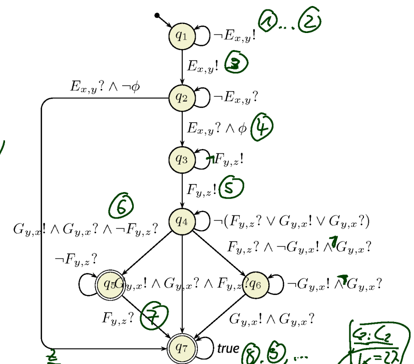
A **set of words** $W \subseteq (Expr_{\mathcal{B}} \to \mathbb{B})^\omega$ is **accepted** by $\mathscr{L}$ if and only if

$\Theta_{\mathscr{L}}$	am = initial	am = invariant
cold	$\exists\beta\,\exists w \in W \bullet w^0 \models_\beta ac \wedge \neg\psi_{exit}(C_0)$ $\wedge\, w^0 \models_\beta \psi_{prog}(\emptyset, C_0) \wedge w/1 \in \mathcal{L}(\mathcal{B}(\mathscr{L}))$	$\exists\beta\,\exists w \in W\, \exists k \in \mathbb{N}_0 \bullet w^k \models_\beta ac \wedge \neg\psi_{exit}(C_0)$ $\wedge\, w^k \models_\beta \psi_{prog}(\emptyset, C_0) \wedge w/k+1 \in \mathcal{L}(\mathcal{B}(\mathscr{L}))$
hot	$\forall\beta\,\forall w \in W \bullet w^0 \models_\beta ac \wedge \neg\psi_{exit}(C_0)$ $\implies w^0 \models_\beta \psi_{prog}(\emptyset, C_0) \wedge w/1 \in \mathcal{L}(\mathcal{B}(\mathscr{L}))$	$\forall\beta\,\forall w \in W\, \forall k \in \mathbb{N}_0 \bullet w^k \models ac \wedge \neg\psi_{exit}(C_0)$ $\implies w^k \models \psi^{\mathsf{Cond}}_{\mathsf{hot}}(\emptyset, C_0) \wedge w/k+1 \in \mathcal{L}(\mathcal{B}(\mathscr{L}))$

where C_0 is the minimal (or **instance heads**) cut.

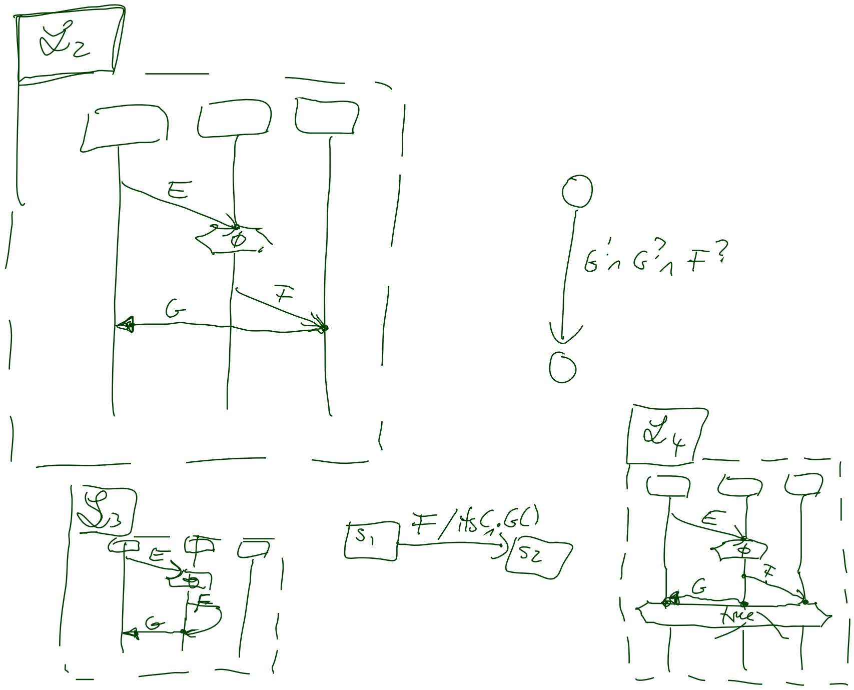
Full LSC Semantics: Example



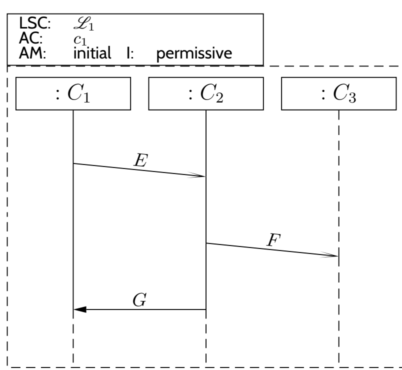


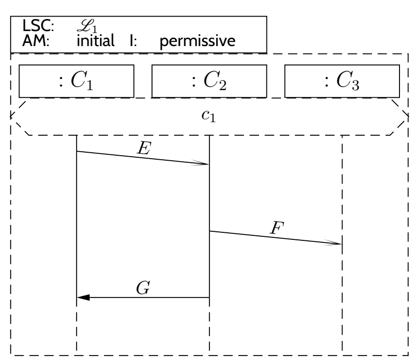
$(\sigma, \varepsilon) \xrightarrow[u]{(cons, Snd)} \cdots \to (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(cons_0, Snd_0)} (\sigma_1, \varepsilon_1) \xrightarrow[c_1]{(cons_1, \{(:E, c_2)\})} (\sigma_2, \varepsilon_2) \xrightarrow[c_2]{(\{:E\}, Snd_2)}$

$(\sigma_3, \varepsilon_3) \xrightarrow[c_2]{(cons_3, \{(:F, c_3)\})} (\sigma_4, \varepsilon_4) \xrightarrow[c_2]{(cons_4, \{(G(), c_1)\})} (\sigma_5, \varepsilon_5) \xrightarrow[c_3]{(\{:F\}, Snd_5)} (\sigma_6, \varepsilon_6) \to \cdots$

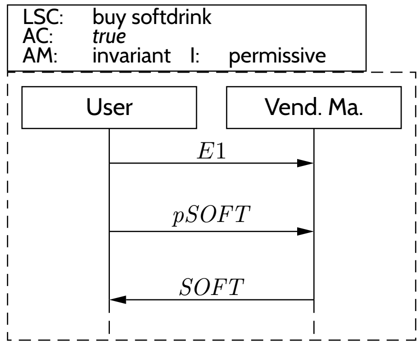


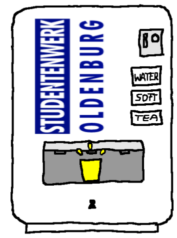
Note: Activation Condition



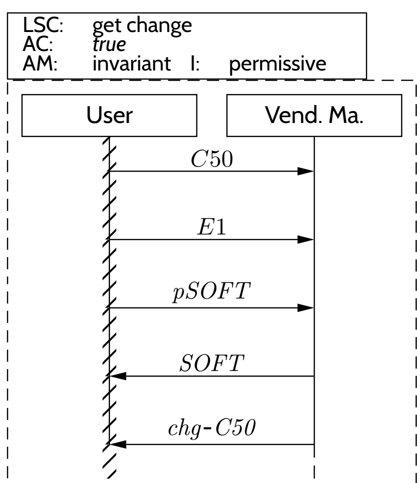


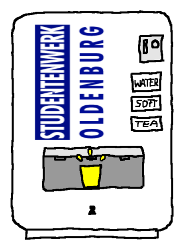
Existential LSC Example: Buy A Softdrink



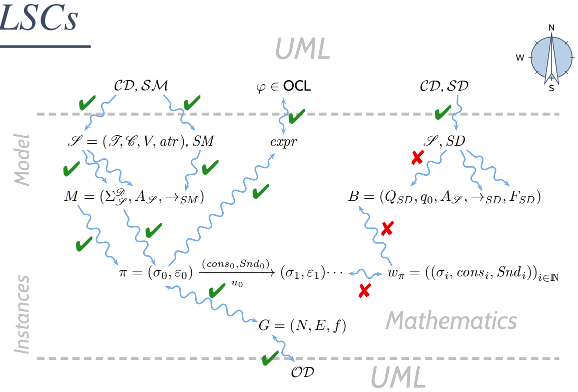


Existential LSC Example: Get Change





TBA-based Semantics of LSCs



Plan:

(i) Given an LSC $\mathscr{L}$ with body

$$((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta),$$

(ii) construct a TBA $\mathcal{B}_{\mathscr{L}}$, and

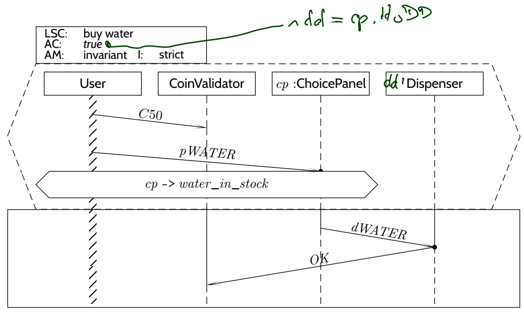
(iii) define language $\mathcal{L}(\mathscr{L})$ of $\mathscr{L}$ **in terms of** $\mathcal{L}(\mathcal{B}_{\mathscr{L}})$,
in particular taking activation condition and activation mode into account.

(iv) define language $\mathcal{L}(\mathcal{M})$ of a UML model.

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

Live Sequence Charts — Precharts

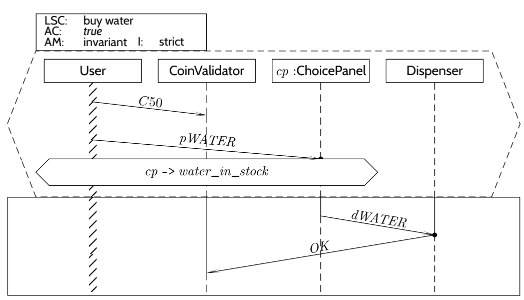
Pre-Charts



A **full LSC** $\mathscr{L} = (PC, MC, ac_0, am, \Theta_{\mathscr{L}})$ **actually** consist of

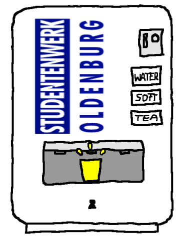
- **pre-chart** $PC = ((L_P, \preceq_P, \sim_P), \mathcal{I}_P, \mathscr{S}, \mathsf{Msg}_P, \mathsf{Cond}_P, \mathsf{LocInv}_P, \Theta_P)$ (possibly empty),
- **main-chart** $MC = ((L_M, \preceq_M, \sim_M), \mathcal{I}_M, \mathscr{S}, \mathsf{Msg}_M, \mathsf{Cond}_M, \mathsf{LocInv}_M, \Theta_M)$ (non-empty),
- **activation condition** $ac_0 : Bool \in Expr_{\mathscr{S}}$,
- **strictness flag** $strict$ (otherwise called **permissive**)
- **activation mode** *am* $\in$ {initial, invariant},
- **chart mode** **existential** ($\Theta_{\mathscr{L}}$ = cold) or **universal** ($\Theta_{\mathscr{L}}$ = hot).

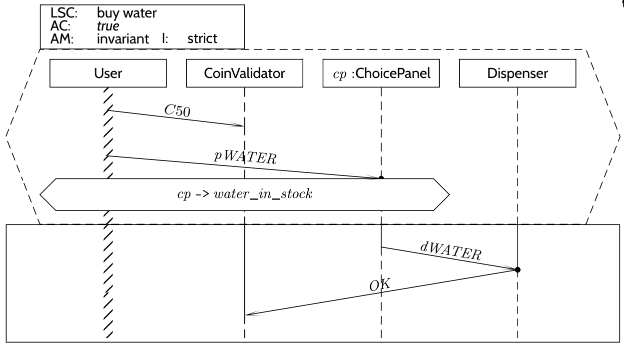
Pre-Charts Semantics



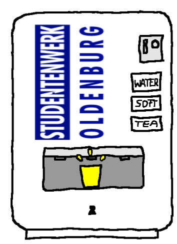
	am = initial	am = invariant
$\Theta_{\mathscr{L}}$ = cold	$\exists\, w \in W\; \exists\, m \in \mathbb{N}_0 \bullet$ $\wedge\, w^0 \models ac \wedge \neg\psi_{exit}(C_0^P) \wedge \psi_{prog}(\emptyset, C_0^P)$ $\wedge\, w^1, \dots, w^m \in \mathcal{L}(\mathcal{B}(PC))$ $\wedge\, w^{m+1} \models \neg\psi_{exit}(C_0^M)$ $\wedge\, w^{m+1} \models \psi_{prog}(\emptyset, C_0^M)$ $\wedge\, w/m+2 \in \mathcal{L}(\mathcal{B}(MC))$	$\exists\, w \in W\; \exists\, k < m \in \mathbb{N}_0 \bullet$ $\wedge\, w^k \models ac \wedge \neg\psi_{exit}(C_0^P) \wedge \psi_{prog}(\emptyset, C_0^P)$ $\wedge\, w^{k+1}, \dots, w^m \in \mathcal{L}(\mathcal{B}(PC))$ $\wedge\, w^{m+1} \models \neg\psi_{exit}(C_0^M)$ $\wedge\, w^{m+1} \models \psi_{prog}(\emptyset, C_0^M)$ $\wedge\, w/m+2 \in \mathcal{L}(\mathcal{B}(MC))$
$\Theta_{\mathscr{L}}$ = hot	$\forall\, w \in W\; \forall\, m \in \mathbb{N}_0 \bullet$ $\wedge\, w^0 \models ac \wedge \neg\psi_{exit}(C_0^P) \wedge \psi_{prog}(\emptyset, C_0^P)$ $\wedge\, w^1, \dots, w^m \in \mathcal{L}(\mathcal{B}(PC))$ $\wedge\, w^{m+1} \models \neg\psi_{exit}(C_0^M)$ $\implies w^{m+1} \models \psi_{prog}(\emptyset, C_0^M)$ $\wedge\, w/m+2 \in \mathcal{L}(\mathcal{B}(MC))$	$\forall\, w \in W\; \forall\, k \leq m \in \mathbb{N}_0 \bullet$ $\wedge\, w^k \models ac \wedge \neg\psi_{exit}(C_0^P) \wedge \psi_{prog}(\emptyset, C_0^P)$ $\wedge\, w^{k+1}, \dots, w^m \in \mathcal{L}(\mathcal{B}(PC))$ $\wedge\, w^{m+1} \models \neg\psi_{exit}(C_0^M)$ $\implies w^{m+1} \models \psi_{prog}(\emptyset, C_0^M)$ $\wedge\, w/m+2 \in \mathcal{L}(\mathcal{B}(MC))$

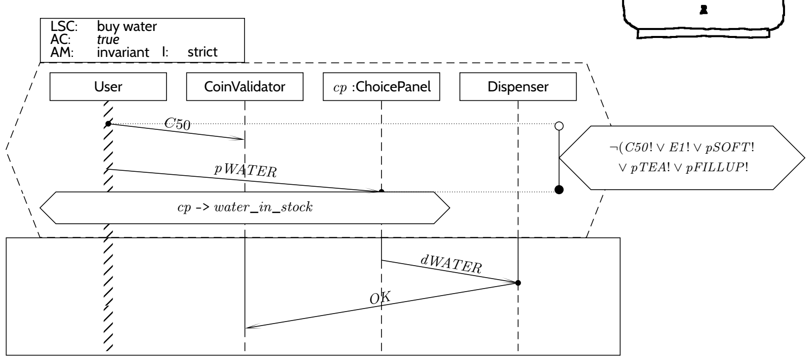
Universal LSC: Example



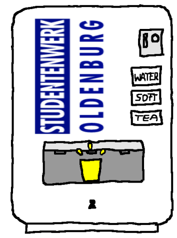


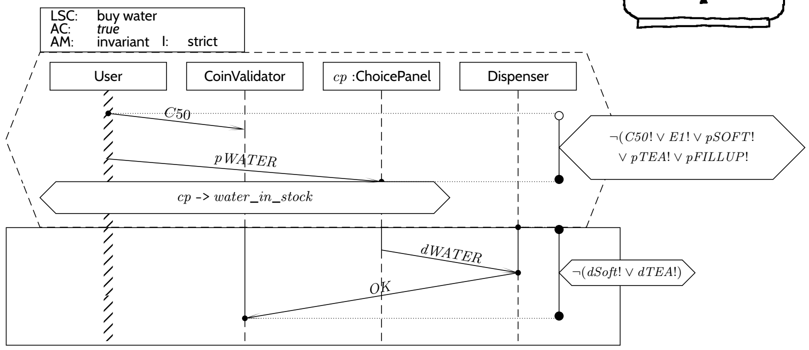
Universal LSC: Example



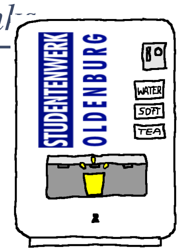


Universal LSC: Example

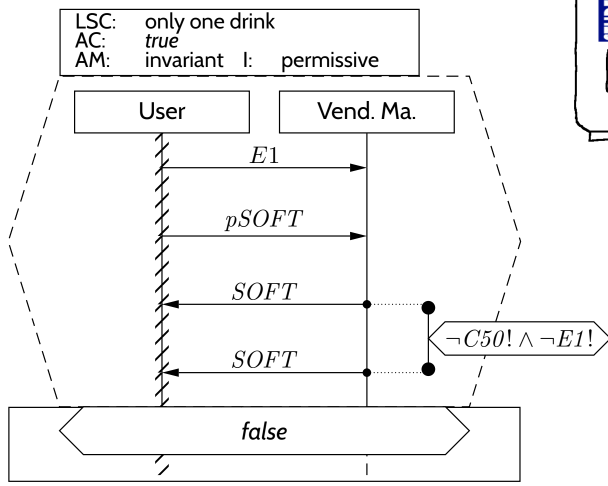


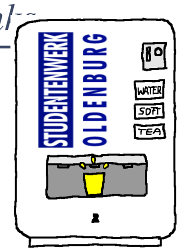


Forbidden Scenario Example: Don't Give Two Drinks

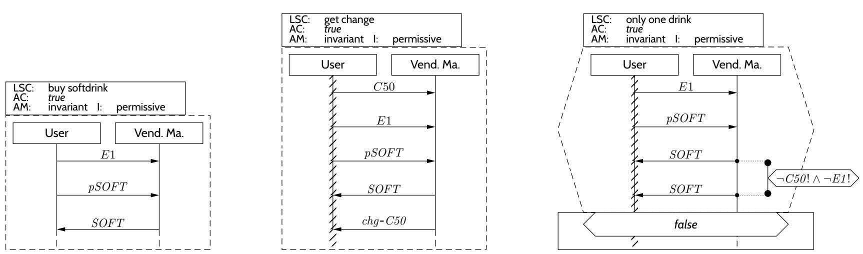


Forbidden Scenario Example: Don't Give Two Drin





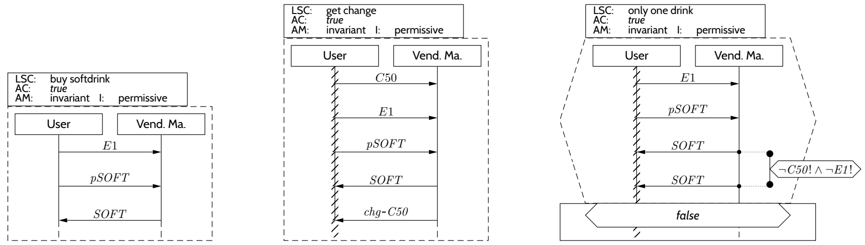
Note: Sequence Diagrams and (Acceptance) Test



- **Existential** LSCs* may hint at **test-cases** for the **acceptance test**!

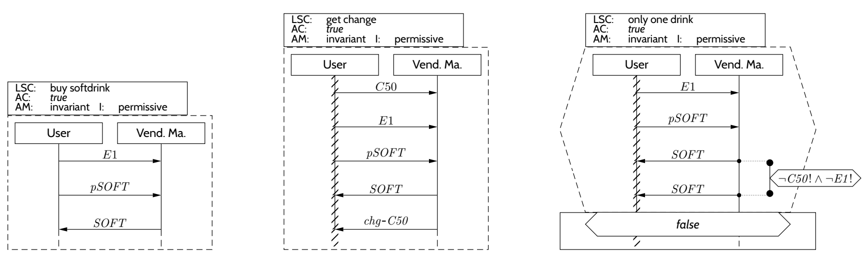
  (∗: as well as (positive) scenarios in general, like use-cases)

Note: Sequence Diagrams 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**!

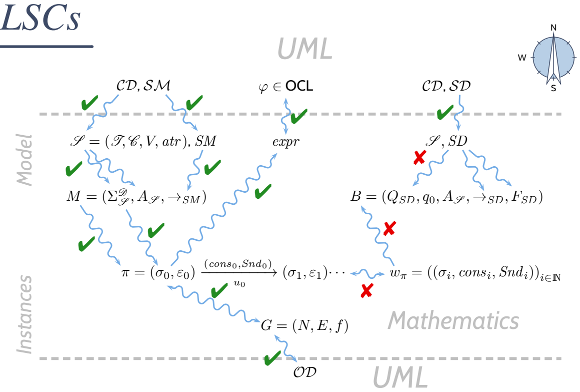
Note: Sequence Diagrams and (Acceptance) Test



- **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.)

TBA-based Semantics of LSCs



Plan:

(i) Given an LSC $\mathscr{L}$ with body

$$((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta),$$

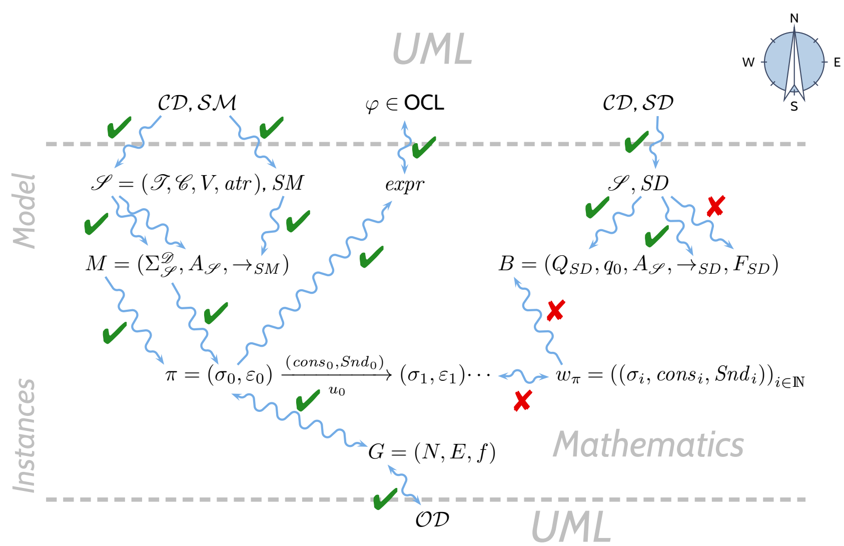
(ii) construct a TBA $\mathcal{B}_{\mathscr{L}}$, and

(iii) define language $\mathcal{L}(\mathscr{L})$ of $\mathscr{L}$ **in terms of** $\mathcal{L}(\mathcal{B}_{\mathscr{L}})$,
in particular taking activation condition and activation mode into account.

(iv) define language $\mathcal{L}(\mathcal{M})$ of a UML model.

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

Course Map



- The **meaning** of an LSC is defined using TBAs.
  - **Cuts** become states of the automaton.
  - Locations induce a **partial order on cuts**.
  - Automaton-transitions and annotations correspond to a **successor relation** on cuts.
  - Annotations use **signal / attribute expressions**.
- **Büchi automata** accept **infinite words**
  - if there **exists is a run** over the word,
  - which visits an accepting state **infinitely often**.
- **The language of a model** is just a rewriting of **computations** into words over an alphabet.
- An LSC **accepts** a word (of a model) if

  Existential: at least on word (of the model) is accepted by the constructed TBA,

  Universion: all words (of the model) are accepted.
- Activation mode **initial** activates at system startup (only), **invariant** with each satisfied activation condition (or pre-chart).

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.

– 20 – 2017-02-02 – main –