Softwaretechnik / Software-Engineering

Lecture 9: Live Sequence Charts

2017-06-19

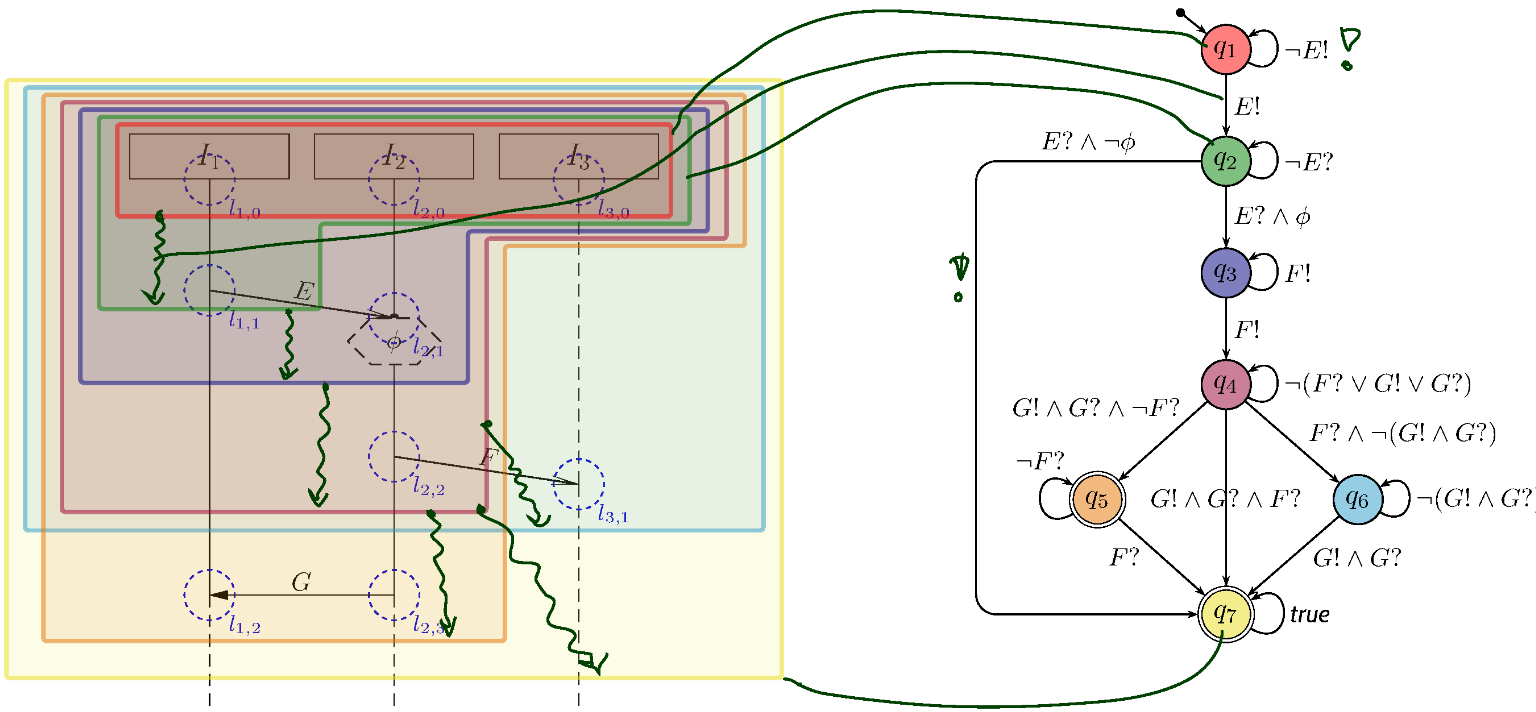
Prof. Dr. Andreas Podelski, **Dr. Bernd Westphal**

Albert-Ludwigs-Universität Freiburg, Germany

Topic Area Requirements Engineering: Content

VL 6	• **Introduction**
	• **Requirements Specification**
	— • Desired Properties
	— • Kinds of Requirements
	— • Analysis Techniques
⋮	• **Documents**
	— • Dictionary, Specification
	• **Specification Languages**
	— • Natural Language
VL 7	— • Decision Tables
	—— • Syntax, Semantics
⋮	—— • Completeness, Consistency, …
VL 8	— • Scenarios
⋮	—— • User Stories, Use Cases
	—— • Live Sequence Charts
VL 9	——— • Syntax, Semantics
⋮	—— • Working Definition: Software
	• **Discussion**

Language of LSC Body: Example



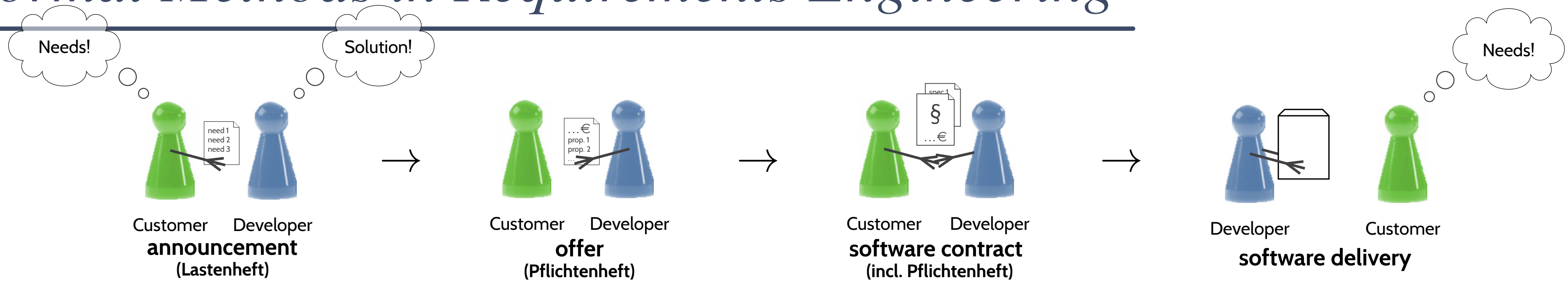
The TBA $\mathcal{B}(\mathscr{L})$ of LSC $\mathscr{L}$ over $\mathcal{C}$ and $\mathcal{E}$ is $(\mathcal{C}_\mathcal{B}, Q, q_{ini}, \to, Q_F)$ with

- $\mathcal{C}_\mathcal{B} = \mathcal{C} \,\dot\cup\, \mathcal{E}_{!?}$, where $\mathcal{E}_{!?} = \{E!, E? \mid E \in \mathcal{E}\}$,
- Q is **the set of cuts** of $\mathscr{L}$, q_{ini} is the **instance heads** cut,
- $\to$ consists of **loops**, **progress transitions** (from $\rightsquigarrow_\mathcal{F}$), 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.

Content

- **Formal Methods in Requirements Engineering**
  - **Software** & **Software Specification**, formally
  - **Requirements Engineering**, formally
  - **Examples**:
    - Decision Tables
    - Use Cases
    - Live Sequence Charts
- **LSC Semantics**:
  - **Full LSC** syntax
  - Activation, Pre-Chart, Chart Mode
- **Automaton Construction**
  - **Loop** / **Progress** / **Exit** Conditions
- **LSCs vs. Software**
- Excursion: **Symbolic Büchi Automata**
- **Methodology**
  - Requirements Engineering with scenarios
  - Strengthening scenarions into requirements
- **Requirements Engineering Wrap-Up**

Formal Methods in Requirements Engineering



- We would like to **precisely** and **objectively** **specify** the **allowed softwares** that make the customer happy.

- In other words, we want to formally define a **satisfies** relation between softwares and software specifications.

  That is, given a software S and a software specification $\mathscr{S}$,
  we want to define when (and only when) software S **satisfies** software specification $\mathscr{S}$,
  denoted by

$$S \models \mathscr{S}.$$

- Once again:
  - $S \models \mathscr{S}$: specification is **satisfied**, S is one "allowed" design, should be accepted.
  - $S \not\models \mathscr{S}$: specification is **not satisfied**, S may not satisfy customer's needs.

Software and Software Specification, formally

Definition. **Software** is a finite description S of a (possibly infinite) set $[\![S]\!]$ of (finite or infinite) computation paths of the form

$$\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots$$

where

- $\sigma_i \in \Sigma$, $i \in \mathbb{N}_0$, is called **state** (or **configuration**), and
- $\alpha_i \in A$, $i \in \mathbb{N}_0$, is called **action** (or **event**).

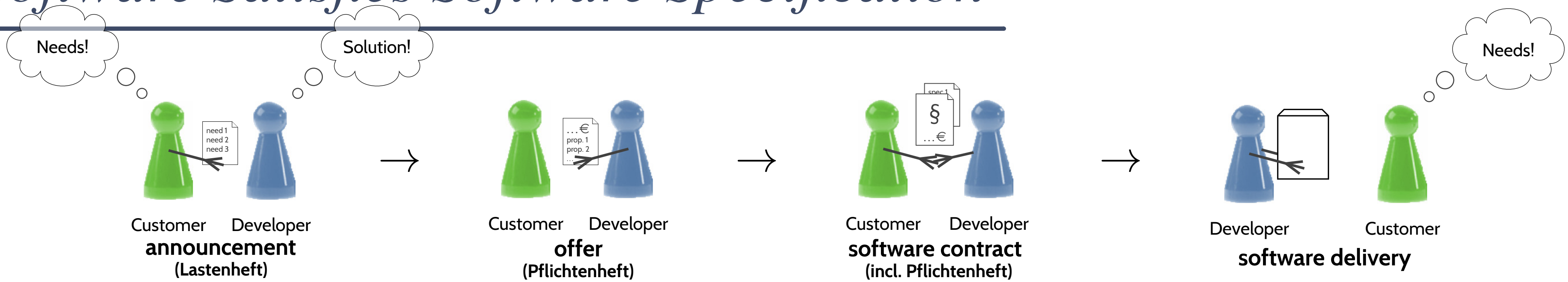
The (possibly partial) function $[\![\,\cdot\,]\!] : S \mapsto [\![S]\!]$ is called **interpretation** of S.

Definition. A **software specification** is a finite description $\mathscr{S}$ of a (possibly infinite) set $[\![\mathscr{S}]\!]$ of softwares, i.e.

$$[\![\mathscr{S}]\!] = \{(S_1, [\![\,\cdot\,]\!]_1), (S_2, [\![\,\cdot\,]\!]_2), \dots\}.$$

The (possibly partial) function $[\![\,\cdot\,]\!] : \mathscr{S} \mapsto [\![\mathscr{S}]\!]$ is called **interpretation** of $\mathscr{S}$.

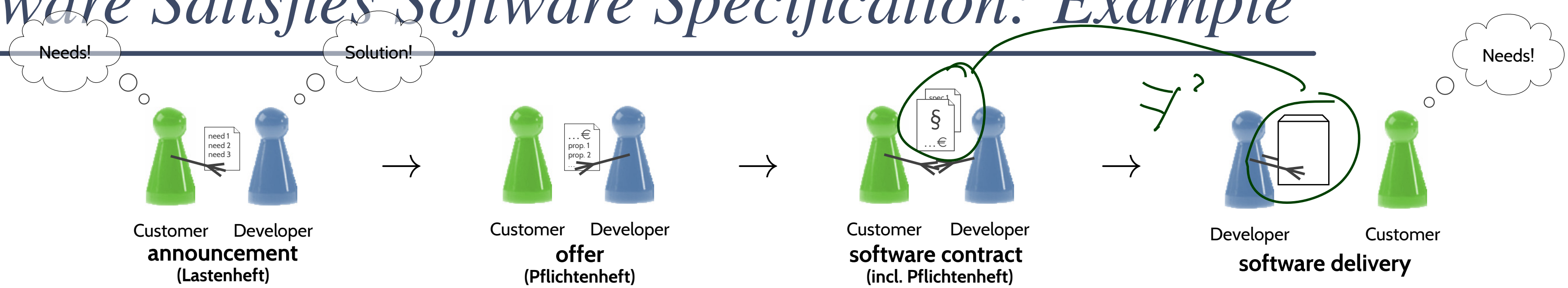
Software Satisfies Software Specification



Definition. Software $(S, [\![\cdot]\!])$ **satisfies** software specification $\mathscr{S}$, denoted by $S \models \mathscr{S}$, if and only if

$$(S, [\![\cdot]\!]) \in [\![\mathscr{S}]\!].$$

Software Satisfies Software Specification: Example



Software Specification

$\mathscr{S}$:

T: room ventilation		r_1	r_2	r_3
b	button pressed?	$\times$	$\times$	$-$
off	ventilation off?	$\times$	$-$	$*$
on	ventilation on?	$-$	$\times$	$*$
go	start ventilation	$\times$	$-$	$-$
$stop$	stop ventilation	$-$	$\times$	$-$

Define: $(S, [\![\cdot]\!]) \in [\![\mathscr{S}]\!]$ if and only if for all

$$\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots \in [\![S]\!]$$

and for all $i \in \mathbb{N}_0$,

$$\exists\, r \in T \bullet \sigma_i \models \mathcal{F}(r).$$

Software

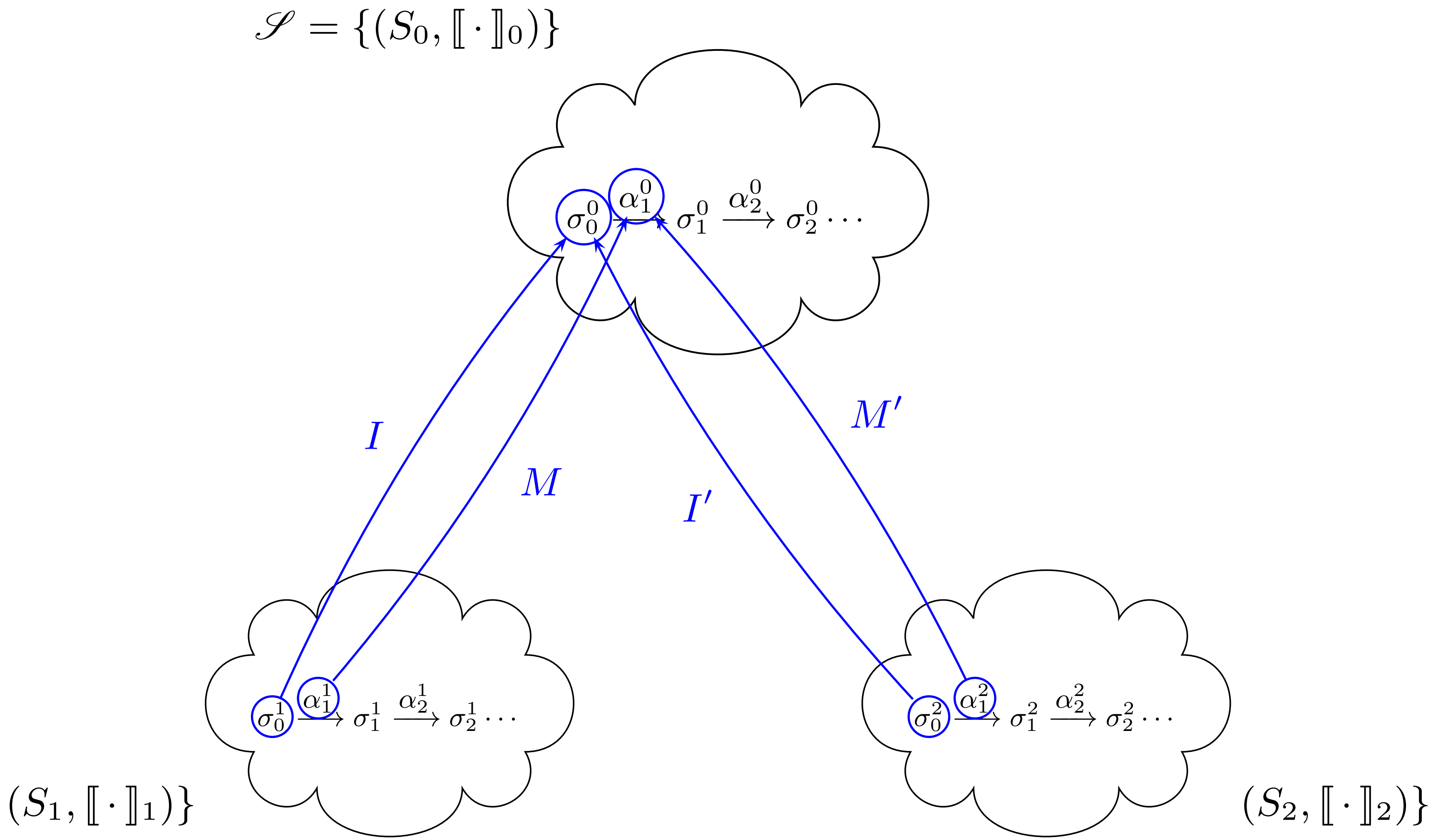
- Assume we have a program S for the room ventilation controller.
- Assume we can **observe** at well-defined points in time the conditions b, off, on, go, $stop$ when the software runs.
- Then **the behaviour** $[\![S]\!]$ of S can be viewed as computation paths of the form

$$\sigma_0 \xrightarrow{\tau} \sigma_1 \xrightarrow{\tau} \sigma_2 \cdots$$

  where each σ_i is a valuation of b, off, on, go, $stop$, i.e. $\sigma_i : \{b, \mathit{off}, on, go, stop\} \to \mathbb{B}$.
- Assume there is $\sigma_0 \xrightarrow{\tau} \sigma_1 \cdots \in [\![S]\!]$ with

$$\sigma_1 = \{b \mapsto 0, \mathit{off} \mapsto 1, on \mapsto 0, go \mapsto 1, stop \mapsto 0\}.$$

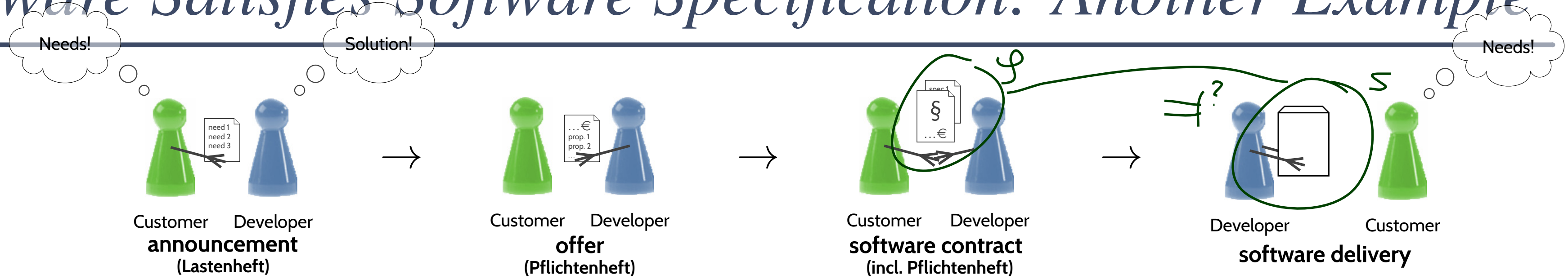
Software Specification vs. Software



S_1 **implements** $\mathscr{S}$
via I and M

S_2 **implements** $\mathscr{S}$
via I and M

Software Satisfies Software Specification: Another Example



Software Specification

$\mathscr{S}$:

- Example positive **scenarios**
- Example negative **scenarios**
- **Use Cases** with pre-condition

Define: $(S, [\![\cdot]\!]) \in [\![\mathscr{S}]\!]$ if and only if

- for each **positive** scenario, there **is a** corresponding $\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots \in [\![S]\!]$,
- for each **negative** scenario, there **is no** corresponding $\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots \in [\![S]\!]$,
- for each **use case** with pre-condition,
  if some σ_i satisfies the pre-condition, then

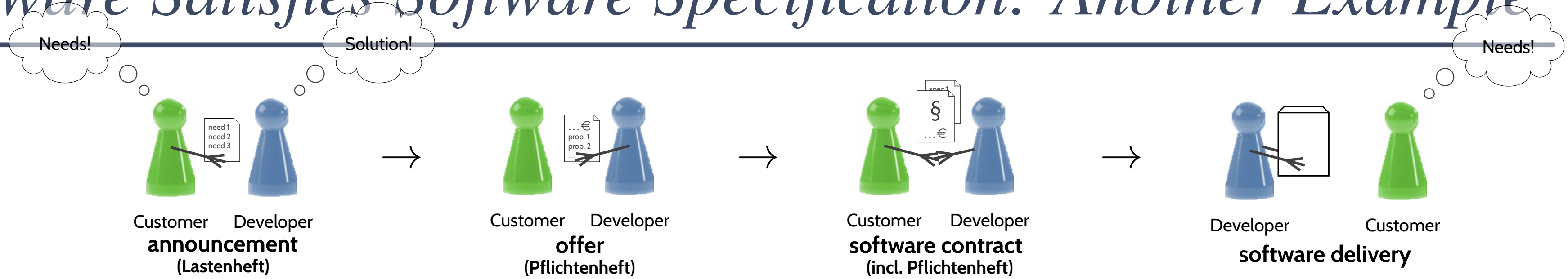
$$\sigma_i \xrightarrow{\alpha_{i+1}} \sigma_{i+1} \xrightarrow{\alpha_{i+2}} \cdots$$

corresponds to the use case.

Software

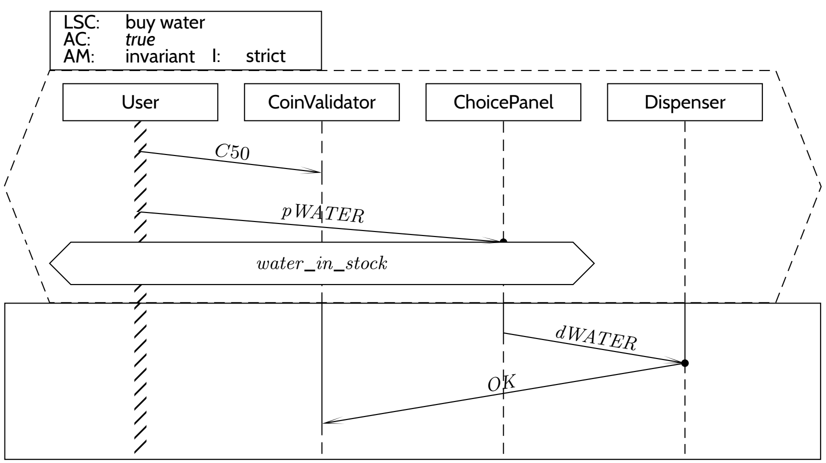
- Assume we can **observe** at well-defined points in time the observables relevant for the use cases when the software S runs.
- Then **the behaviour** $[\![S]\!]$ of S can be viewed as computation paths where each state σ_i is a valuation of the use case's observables.
- And then we can relate S to $\mathscr{S}$.

Software Satisfies Software Specification: Another Example



Software Specification

$\mathscr{S}$:



Define: $(S, [\![\cdot]\!]) \in [\![\mathscr{S}]\!]$ if and only if

- **tja...** (in a minute)

Software

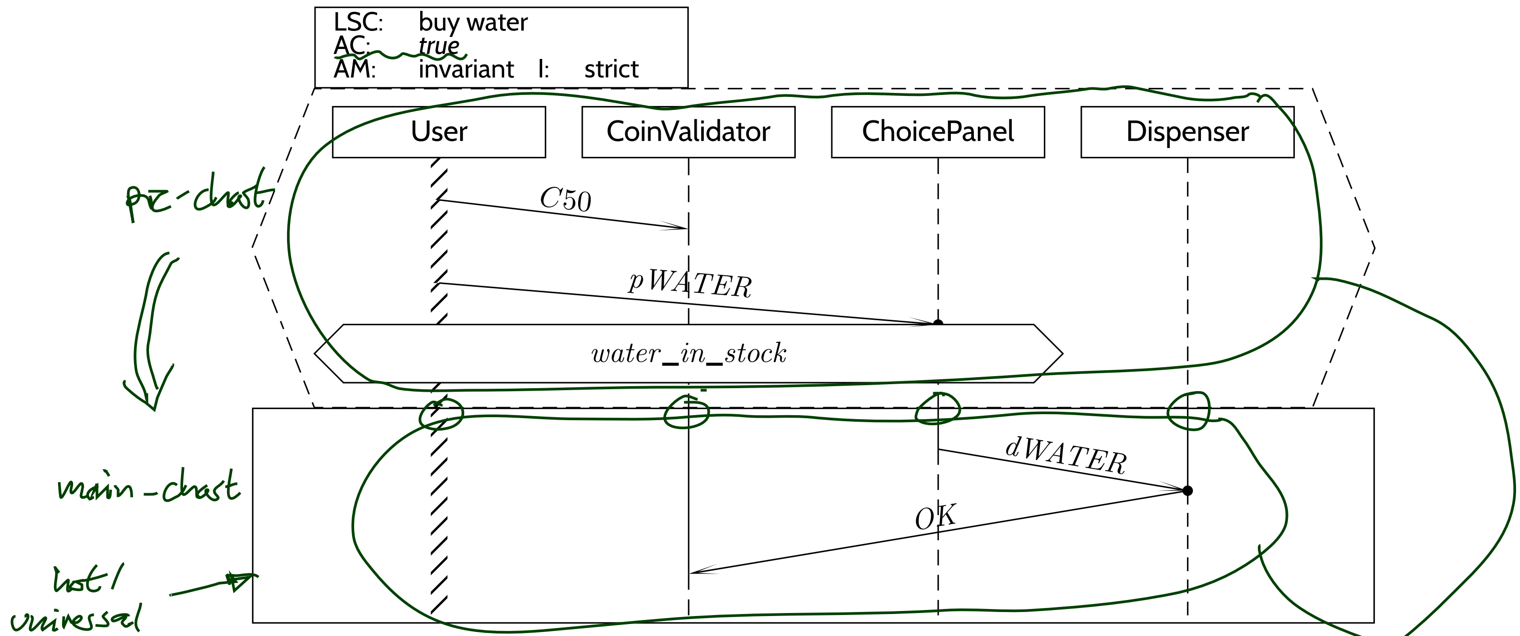
- Assume we can **observe** at well-defined points in time the observables relevant for the LSC (conditions and messages) when the software S runs.
- Then **the behaviour** $[\![S]\!]$ of S can be viewed as computation paths over the LSC's observables.
- And then we can relate S to $\mathscr{S}$.

Content

- **Formal Methods in Requirements Engineering**
  - **Software** & **Software Specification**, formally
  - **Requirements Engineering**, formally
  - **Examples**:
    - Decision Tables
    - Use Cases
    - Live Sequence Charts
- **LSC Semantics**:
  - **Full LSC** syntax
  - Activation, Pre-Chart, Chart Mode
- **Automaton Construction**
  - **Loop** / **Progress** / **Exit** Conditions
- **LSCs vs. Software**
- Excursion: **Symbolic Büchi Automata**
- **Methodology**
  - Requirements Engineering with scenarios
  - Strengthening scenarions into requirements
- **Requirements Engineering Wrap-Up**

LSC Semantics

Full LSC Syntax

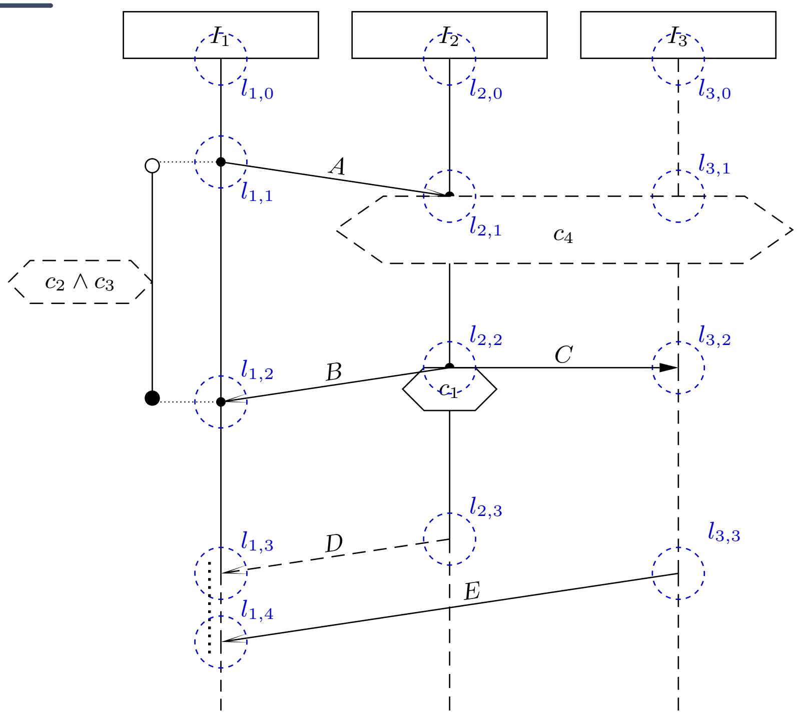


A **full LSC** $\mathscr{L} = (PC, MC, ac_0, am, \Theta_{\mathscr{L}})$ **actually** consist of

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

From Concrete to Abstract Syntax

- locations $\mathcal{L}$,
- $\preceq \subseteq \mathcal{L} \times \mathcal{L}$, $\quad \sim \subseteq \mathcal{L} \times \mathcal{L}$
- $\mathcal{I} = \{I_1, \dots, I_n\}$,
- Msg $\subseteq \mathcal{L} \times \mathcal{E} \times \mathcal{L}$,
- Cond $\subseteq (2^{\mathcal{L}} \setminus \emptyset) \times \Phi(\mathcal{C})$
- LocInv $\subseteq \mathcal{L} \times \{\circ, \bullet\} \times \Phi(\mathcal{C}) \times \mathcal{L} \times \{\circ, \bullet\}$,
- Θ : $\mathcal{L} \cup$ Msg $\cup$ Cond $\cup$ LocInv $\to$ {hot, cold}.



- $\mathcal{L} = \{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, \; l_{1,2}, l_{1,4}, \; l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}, \; l_{3,0}, l_{3,1}, l_{3,2}, l_{3,3}\}$
- $l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2} \prec l_{3,3},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,1} \sim l_{3,1}, \quad l_{2,2} \sim l_{3,2},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$ $l_{3,3}$
- Msg $= \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,2}), (l_{2,3}, D, l_{1,3}), (l_{3,3}, E, l_{1,4})\}$
- Cond $= \{(\{l_{2,1}, l_{3,1}\}, c_4), (\{l_{2,2}\}, c_2 \wedge c_3)\}$,
- LocInv $= \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$

LSC Semantics

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

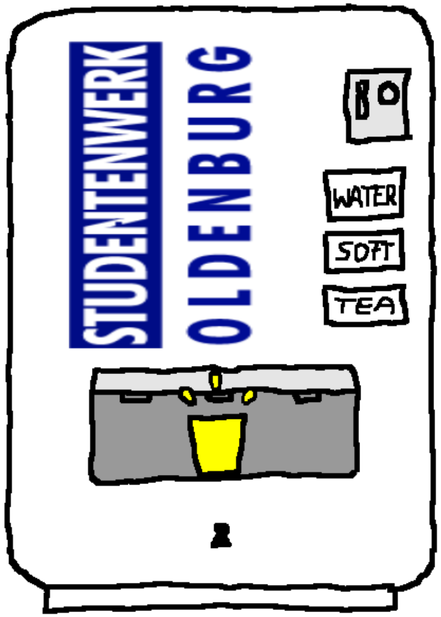
- Here, W is a set of words (for the moment, think of computation paths, like $[\![S]\!]$).
- $w \in W$ is a word (for the moment, think of a computation path, like $\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots \in [\![S]\!]$).

LSC Semantics

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

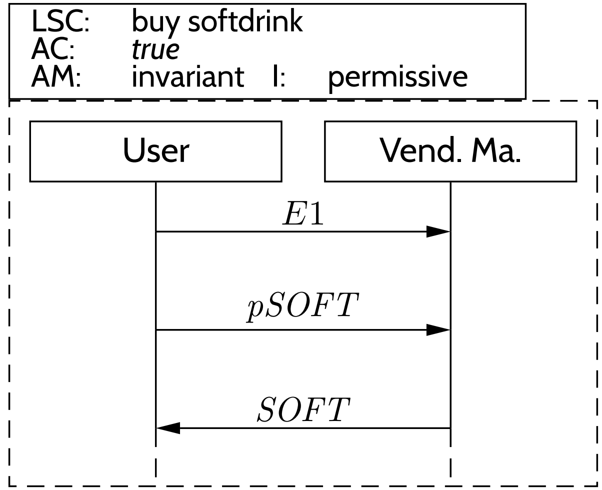
- Here W is a set of words (for the moment, think of computation paths, like $[\![S]\!]$).
- $w \in W$ is a word (for the moment, think of a computation path, like $\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots \in [\![S]\!]$).

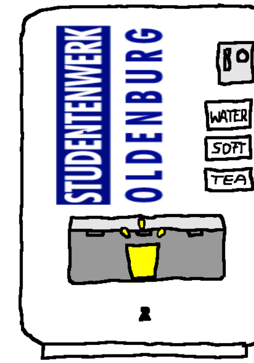
Example: Vending Machine



- **Positive scenario**: Buy a Softdrink
  (i) Insert one 1 euro coin.
  (ii) Press the 'softdrink' button.
  (iii) Get a softdrink.
- **Positive scenario**: Get Change
  (i) Insert one 50 cent and one 1 euro coin.
  (ii) Press the 'softdrink' button.
  (iii) Get a softdrink.
  (iv) Get 50 cent change.
- **Negative scenario**: A Drink for Free
  (i) Insert one 1 euro coin.
  (ii) Press the 'softdrink' button.
  (iii) Do not insert any more money.
  (iv) Get **two** softdrinks.

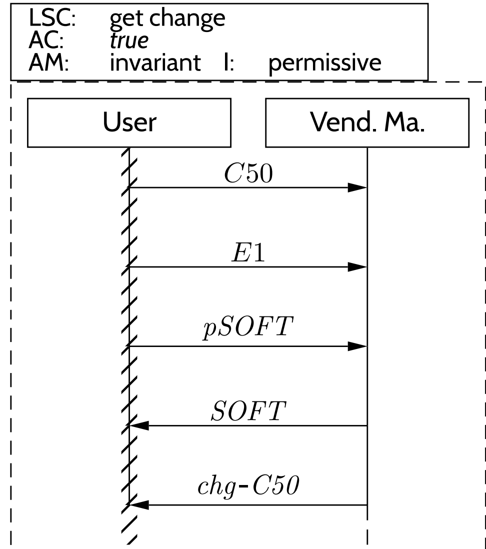
LSC Semantics





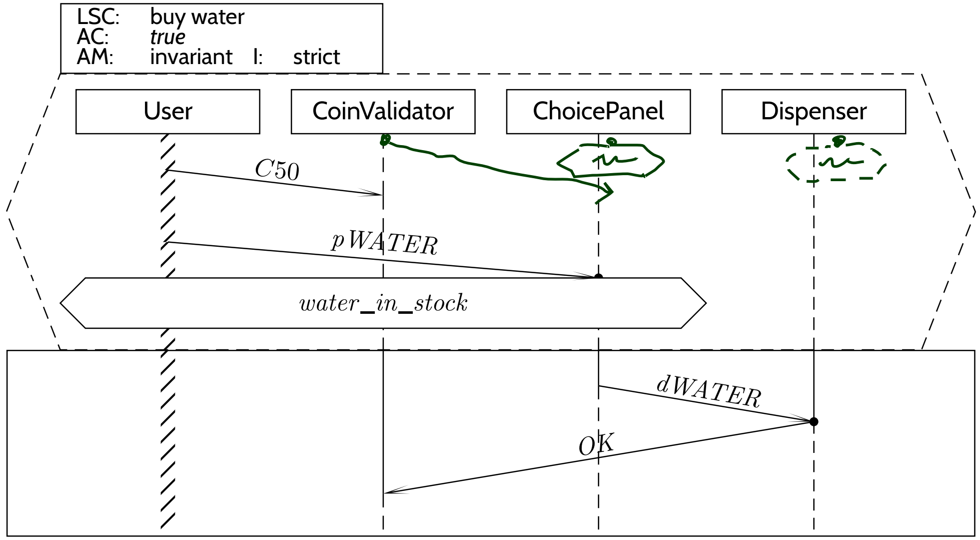
	$am =$ initial	$am =$ invariant
$\Theta_\mathscr{L} =$ cold	$\exists\, w \in W \;\exists\, m \in \mathbb{N}_0 \bullet$ $\wedge\, w^0 \models ac \wedge \neg\psi_{exit}(C_0^P) \wedge \psi_{prog}(\emptyset, C_0^P)$ $\wedge\, w/1, \dots, w/m \in Lang(\mathcal{B}(PC))$ $\wedge\, w^{m+1} \models \neg\psi_{exit}(C_0^M)$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

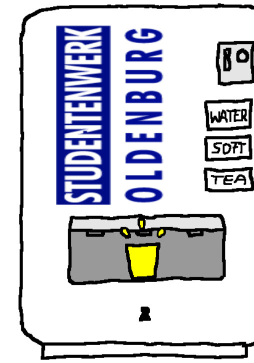
LSC Semantics



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

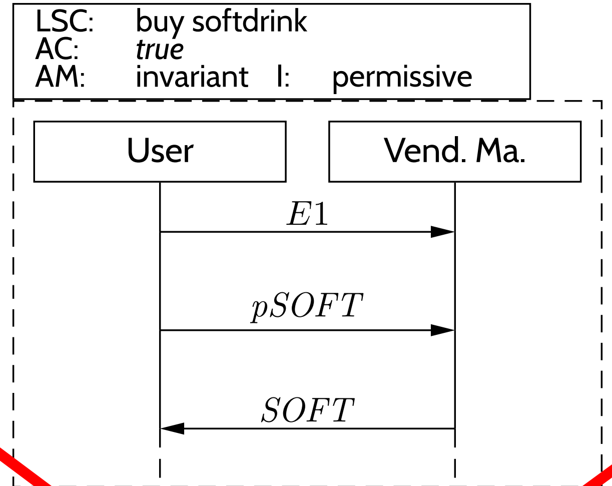
LSC Semantics

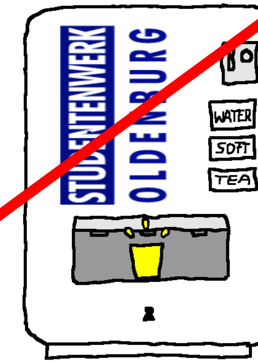




	am = initial	am = invariant
$\Theta_{\mathscr{L}}$ = cold	$\exists\, w \in W\ \exists\, m \in \mathbb{N}_0 \bullet$ $\wedge\ w^0 \models ac \wedge \neg\psi_{exit}(C_0^P) \wedge \psi_{prog}(\emptyset, C_0^P)$ $\wedge\ w/1, \dots, w/m \in Lang(\mathcal{B}(PC))$ $\wedge\ w^{m+1} \models \neg\psi_{exit}(C_0^M)$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

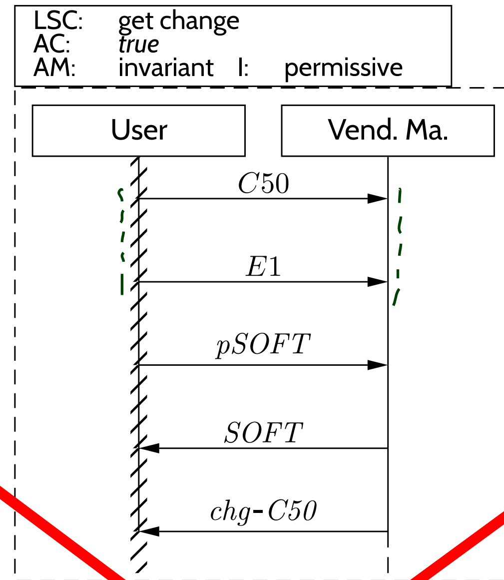
LSC Semantics

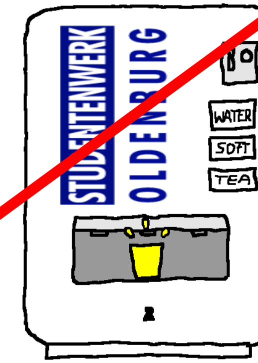




$\Theta_{\mathscr{L}}$	$am = \text{initial}$	$am = \text{invariant}$
cold	$\exists\, w \in W\ \exists\, m \in \mathbb{N}_0 \bullet w^0 \models ac$ $\wedge\ w^0 \models \psi_{\text{hot}}^{\text{Cond}}(\emptyset, C_0^P)$ $\wedge\ w/1, \dots, w/m \in Lang(\mathcal{B}(PC))$ $\wedge\ w^{m+1} \models \psi_{\text{hot}}^{\text{Cond}}(\emptyset,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

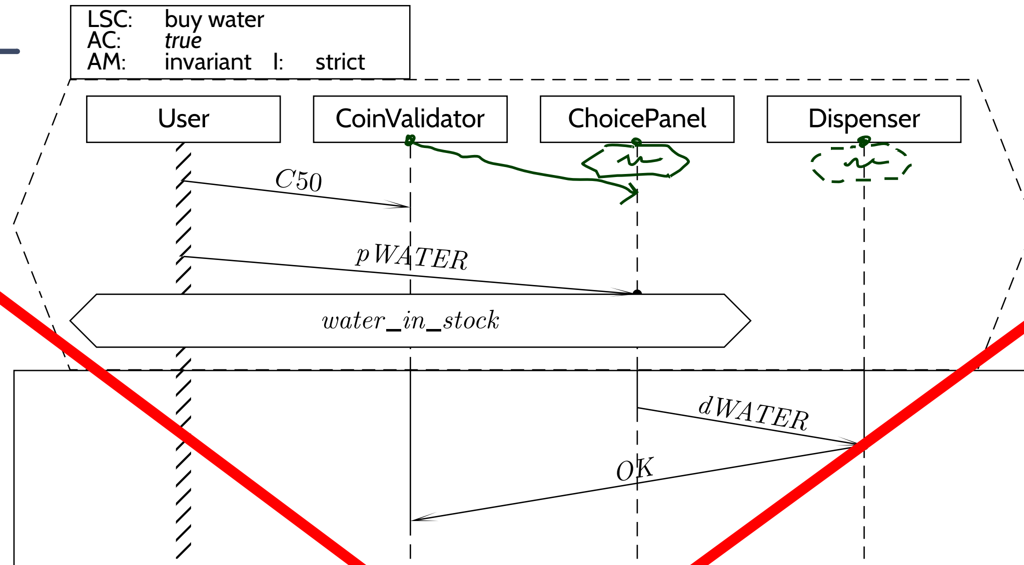
LSC Semantics

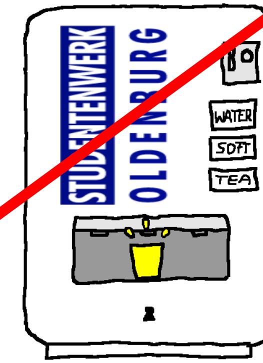




$\Theta_{\mathscr{L}}$	am = initial	am = invariant
cold	$\exists\, w \in W\ \exists\, m \in \mathbb{N}_0 \bullet w^0 \models ac$ $\wedge\ w^0 \models \psi^{\mathsf{Cond}}_{\mathsf{hot}}(\emptyset, C_0^P)$ $\wedge\ w/1, \dots, w/m \in Lang(\mathcal{B}(PC))$ $\wedge\ w^{m+1} \models \psi^{\mathsf{Cond}}_{\mathsf{hot}}(\emptyset,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

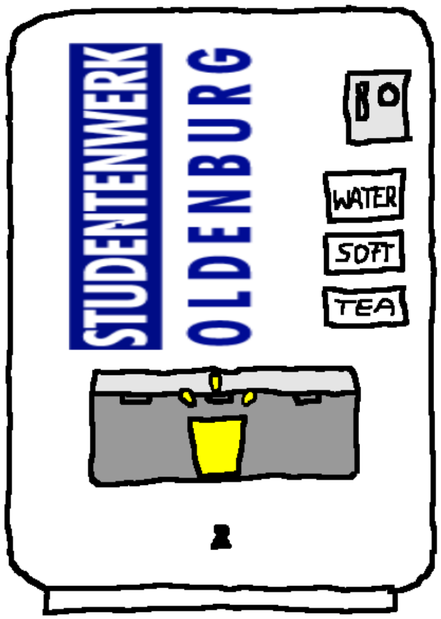
LSC Semantics





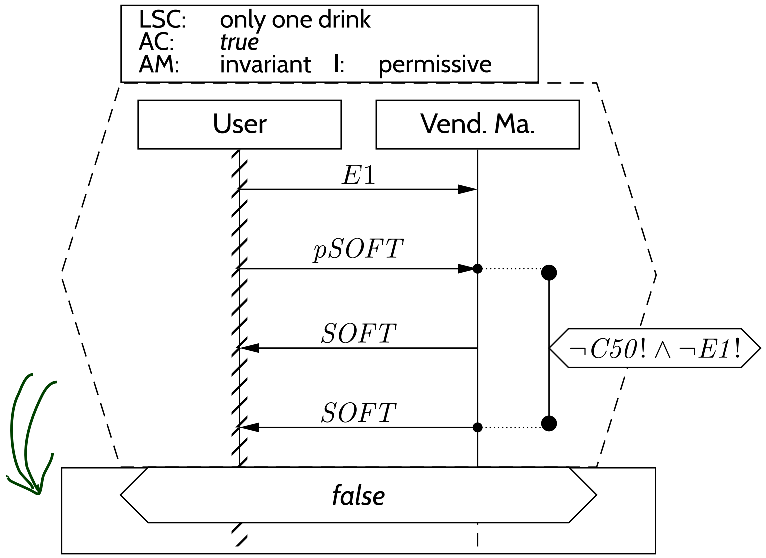
$\Theta_{\mathscr{L}}$	$am =$ initial	$am =$ invariant
cold	$\exists\, w \in W\, \exists\, m \in \mathbb{N}_0 \bullet w^0 \models ac$ $\wedge\, w^0 \models \psi_{\text{hot}}^{\text{Cond}}(\emptyset, C_0^P)$ $\wedge\, w/1, \dots, w/m \in Lang(\mathcal{B}(PC))$ $\wedge\, w^{m+1} \models \psi_{\text{hot}}^{\text{Cond}}(\emptyset,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

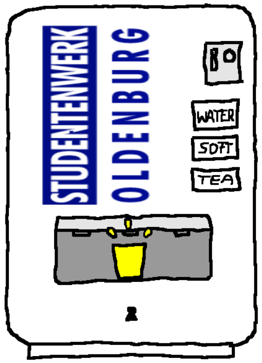
Example: Vending Machine



- **Positive scenario**: Buy a Softdrink
  (i) Insert one 1 euro coin.
  (ii) Press the 'softdrink' button.
  (iii) Get a softdrink.
- **Positive scenario**: Get Change
  (i) Insert one 50 cent and one 1 euro coin.
  (ii) Press the 'softdrink' button.
  (iii) Get a softdrink.
  (iv) Get 50 cent change.
- **Negative scenario**: A Drink for Free
  (i) Insert one 1 euro coin.
  (ii) Press the 'softdrink' button.
  (iii) Do not insert any more money.
  (iv) Get **two** softdrinks.

LSC Semantics

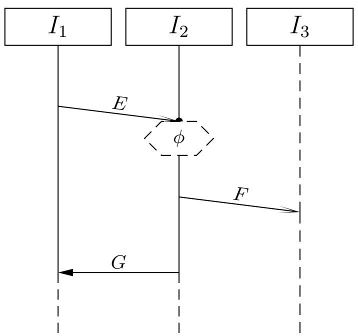




$\Theta_{\mathscr{L}}$	am = initial	am = invariant
cold	$\exists\, w \in W\ \exists\, m \in \mathbb{N}_0 \bullet w^0 \models ac$ $\wedge\ w^0 \models \psi_{\text{hot}}^{\text{Cond}}(\emptyset, C_0^P)$ $\wedge\ w/1, \dots, w/m \in Lang(\mathcal{B}(PC))$ $\wedge\ w^{m+1} \models \psi_{\text{hot}}^{\text{Cond}}(\emptyset,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

LSC Semantics: TBA Construction

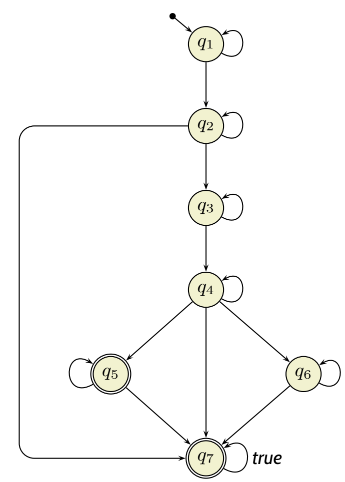
The Plan: A Formal Semantics for a Visual Formalism



concrete syntax
(diagram)

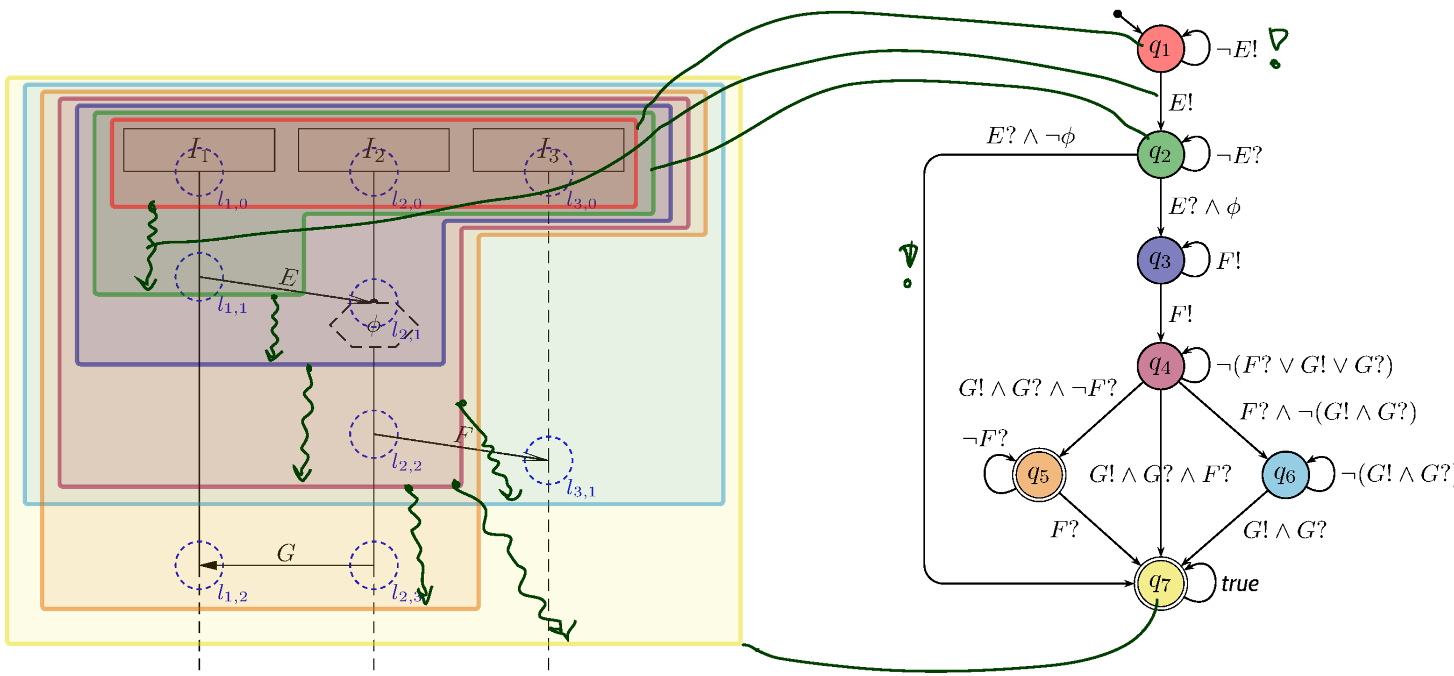
$((\mathcal{L}, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta)$

abstract syntax



semantics
(Büchi automaton)

Language of LSC Body: Example



The TBA $\mathcal{B}(\mathscr{L})$ of LSC $\mathscr{L}$ over $\mathcal{C}$ and $\mathcal{E}$ is $(\mathcal{C}_\mathcal{B}, Q, q_{ini}, \rightarrow, Q_F)$ with

- $\mathcal{C}_\mathcal{B} = \mathcal{C} \,\dot\cup\, \mathcal{E}_{!?}$, where $\mathcal{E}_{!?} = \{E!, E? \mid E \in \mathcal{E}\}$,
- Q is **the set of cuts** of $\mathscr{L}$, q_{ini} is the **instance heads** cut,
- $\rightarrow$ consists of **loops**, **progress transitions** (from $\rightsquigarrow_\mathcal{F}$), and **legal exits** (cold cond./local inv.),
- $Q_F = \{C \in Q \mid \Theta(C) = \text{cold} \vee C = \mathcal{L}\}$ is the set of cold cuts and the maximal cut.

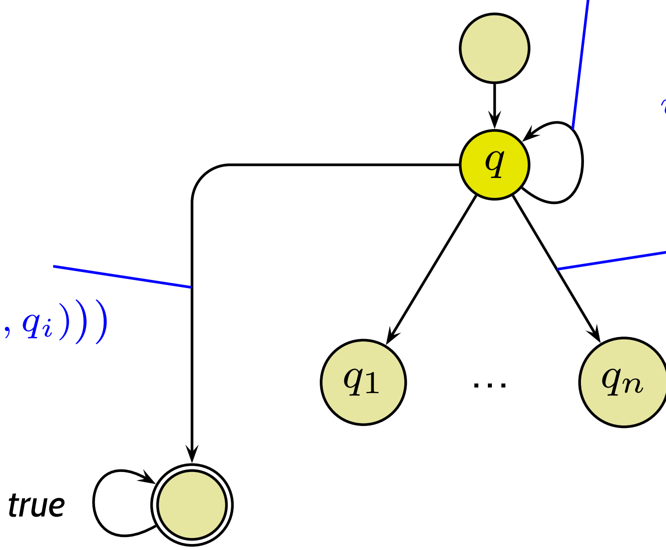
TBA Construction Principle

"Only" construct the transitions' labels:

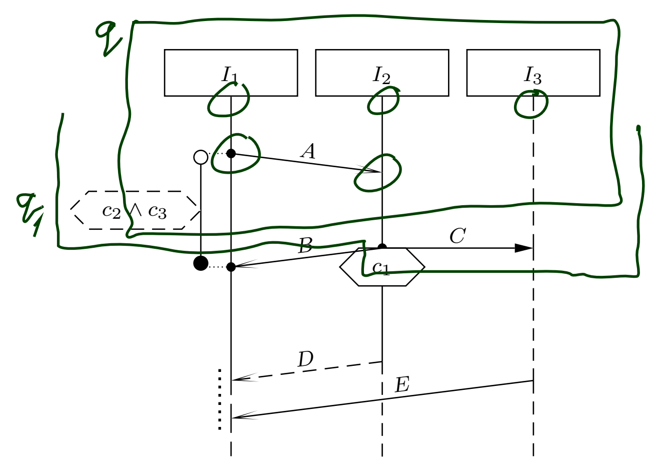
$$\to = \{(q, \psi_{loop}(q), q) \mid q \in Q\} \cup \{(q, \psi_{prog}(q, q'), q') \mid q \rightsquigarrow_{\mathcal{F}} q'\} \cup \{(q, \psi_{exit}(q), \mathcal{L}) \mid q \in Q\}$$

$$\psi_{loop}(q) = \overbrace{\psi^{\mathsf{Msg}}(q) \wedge \psi^{\mathsf{LocInv}}_{\mathsf{hot}}(q)}^{=:\psi^{\mathsf{hot}}_{loop}(q)} \wedge \psi^{\mathsf{LocInv}}_{\mathsf{cold}}(q)$$

$$\psi_{exit}(q) =$$
$$\left(\psi^{\mathsf{hot}}_{loop}(q) \wedge \neg\psi^{\mathsf{LocInv}}_{\mathsf{cold}}(q)\right)$$
$$\vee \bigvee_{1 \leq i \leq n} \left(\psi^{\mathsf{hot}}_{prog}(q, q_i)\right.$$
$$\left.\wedge \left(\neg\psi^{\mathsf{LocInv},\bullet}_{\mathsf{cold}}(q, q_i) \vee \neg\psi^{\mathsf{Cond}}_{\mathsf{cold}}(q, q_i)\right)\right)$$



$$\psi_{prog}(q, q_n) = \overbrace{\psi^{\mathsf{Msg}}(q, q_n) \wedge \psi^{\mathsf{Cond}}_{\mathsf{hot}}(q, q_n) \wedge \psi^{\mathsf{LocInv},\bullet}_{\mathsf{hot}}(q, q_n)}^{=:\psi^{\mathsf{hot}}_{prog}(q, q_n)}$$
$$\wedge\, \psi^{\mathsf{Cond}}_{\mathsf{cold}}(q, q_n) \wedge \psi^{\mathsf{LocInv},\bullet}_{\mathsf{cold}}(q, q_n)$$



Loop Condition

$$\psi_{loop}(q) = \psi^{\mathsf{Msg}}(q) \wedge \psi^{\mathsf{LocInv}}_{\mathsf{hot}}(q) \wedge \psi^{\mathsf{LocInv}}_{\mathsf{cold}}(q)$$

- $\psi^{\mathsf{Msg}}(q) = \neg \bigvee_{1 \leq i \leq n} \psi^{\mathsf{Msg}}(q, q_i) \wedge \underbrace{\left(strict \implies \bigwedge_{\psi \in \mathcal{E}_{!?} \cap \mathsf{Msg}(\mathcal{L})} \neg\psi\right)}_{=:\psi_{\mathsf{strict}}(q)}$

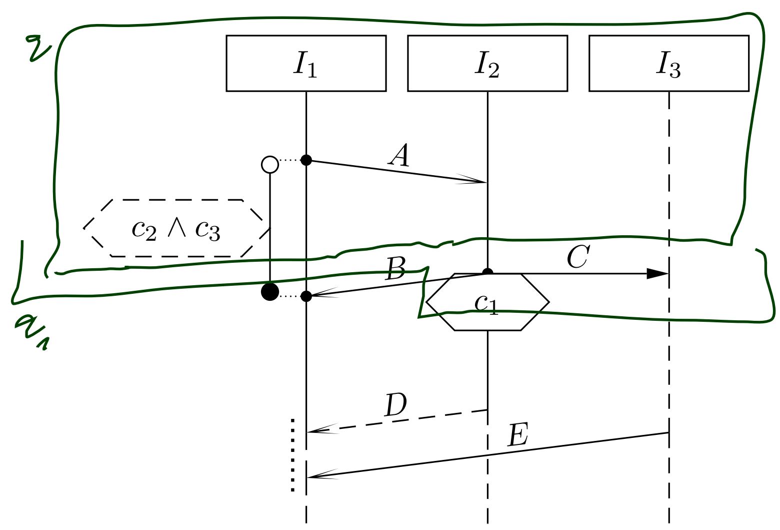
- $\psi^{\mathsf{LocInv}}_{\theta}(q) = \bigwedge_{\ell=(l,\iota,\phi,l',\iota') \in \mathsf{LocInv},\ \Theta(\ell)=\theta,\ \ell \text{ active at } q} \phi$

  A location l is called **front location** of cut C if and only if $\nexists\, l' \in \mathcal{L} \bullet l \prec l'$.

  Local invariant $(l_o, \iota_0, \phi, l_1, \iota_1)$ is **active** at cut (!) q
  if and only if $l_0 \preceq l \prec l_1$ for some front location l of cut q or $l = l_1 \wedge \iota_1 = \bullet$.

- $\mathsf{Msg}(\mathcal{F}) = \{E! \mid (l, E, l') \in \mathsf{Msg},\ l \in \mathcal{F}\} \cup \{E? \mid (l, E, l') \in \mathsf{Msg},\ l' \in \mathcal{F}\}$

- $\mathsf{Msg}(\mathcal{F}_1, \dots, \mathcal{F}_n) = \bigcup_{1 \leq i \leq n} \mathsf{Msg}(\mathcal{F}_i)$



Progress Condition

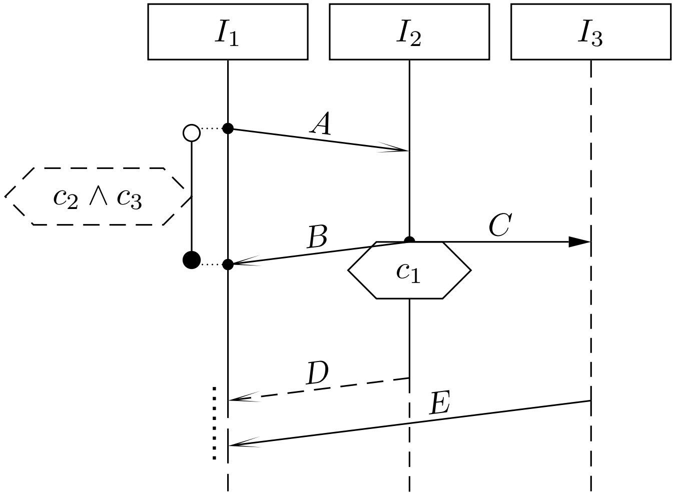
$$\psi^{\mathsf{hot}}_{prog}(q, q_i) = \psi^{\mathsf{Msg}}(q, q_n) \wedge \psi^{\mathsf{Cond}}_{\mathsf{hot}}(q, q_n) \wedge \psi^{\mathsf{LocInv},\bullet}_{\mathsf{hot}}(q_n)$$

- $\psi^{\mathsf{Msg}}(q, q_i) = \bigwedge_{\psi \in \mathsf{Msg}(q_i \setminus q)} \psi \wedge \bigwedge_{j \neq i} \bigwedge_{\psi \in (\mathsf{Msg}(q_j \setminus q) \setminus \mathsf{Msg}(q_i \setminus q))} \neg\psi$
  $\wedge \underbrace{\left(strict \implies \bigwedge_{\psi \in (\mathcal{E}_{!?} \cap \mathsf{Msg}(\mathcal{L})) \setminus \mathsf{Msg}(\mathcal{F}_i)} \neg\psi\right)}_{=:\psi_{\mathsf{strict}}(q, q_i)}$
- $\psi^{\mathsf{Cond}}_{\theta}(q, q_i) = \bigwedge_{\gamma = (L, \phi) \in \mathsf{Cond},\ \Theta(\gamma) = \theta,\ L \cap (q_i \setminus q) \neq \emptyset} \phi$
- $\psi^{\mathsf{LocInv},\bullet}_{\theta}(q, q_i) = \bigwedge_{\lambda = (l, \iota, \phi, l', \iota') \in \mathsf{LocInv},\ \Theta(\lambda) = \theta,\ \lambda\ \bullet\text{-active at } q_i} \phi$

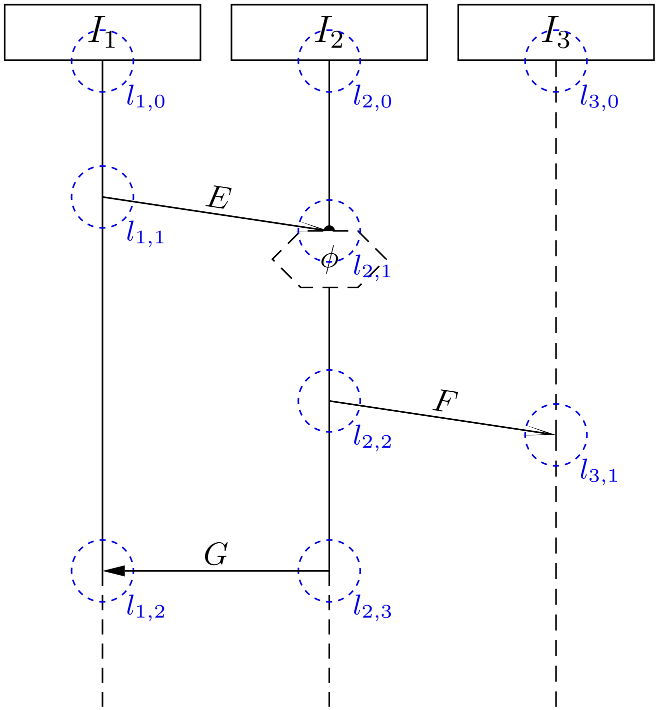
  Local invariant $(l_0, \iota_0, \phi, l_1, \iota_1)$ is **$\bullet$-active** at q if and only if

  - $l_0 \prec l \prec l_1$, or
  - $l = l_0 \wedge \iota_0 = \bullet$, or
  - $l = l_1 \wedge \iota_1 = \bullet$

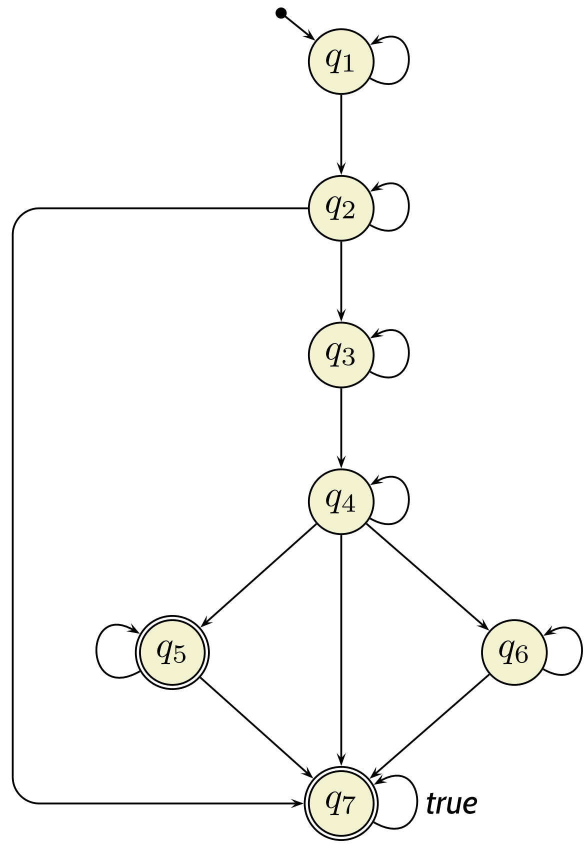
  for some front location l of cut (!) q.



Example



I_1
I_2
I_3
$l_{1,0}$
$l_{2,0}$
$l_{3,0}$
E
$l_{1,1}$
ϕ
$l_{2,1}$
F
$l_{2,2}$
$l_{3,1}$
G
$l_{1,2}$
$l_{2,3}$



q_1
q_2
q_3
q_4
q_5
q_6
q_7
true

Tell Them What You've Told Them...

- **Live Sequence Charts** (if well-formed)
  - have an abstract syntax.
- From an abstract syntax, mechanically construct its **TBA**.
- A **universal** **LSC** is **satisfied** by a software S if and only if
  - **all words** induced by the computation paths of S
  - are **accepted** by the LSC's TBA.
- An **existential** **LSC** is **satisfied** by a software S if and only if
  - **there is a word** induced by a computation path of S
  - which is **accepted** by the LSC's TBA.
- **Pre-charts** allow us to specify
  - anti-scenarios ("this must not happen"),
  - activation interactions.

References

References

Harel, D. and Marelly, R. (2003). *Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine*. Springer-Verlag.

Ludewig, J. and Lichter, H. (2013). *Software Engineering*. dpunkt.verlag, 3. edition.

Rupp, C. and die SOPHISTen (2014). *Requirements-Engineering und -Management*. Hanser, 6th edition.