

Software Design, Modelling and Analysis in UML

Lecture 20: Live Sequence Charts IV

2017-02-02

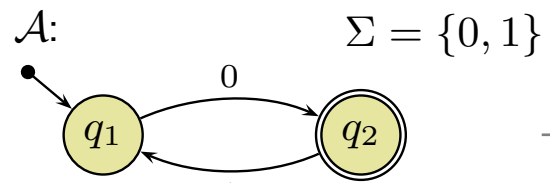
Prof. Dr. Andreas Podelski, Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

- **Live Sequence Charts**
 - **Semantics**
 - **Excursion: Büchi Automata**
 - **Language of a Model**
 - **Full LSCs**
 - **Existential and Universal**
 - **Pre-Charts**
 - **Forbidden Scenarios**
 - **LSCs and Tests**

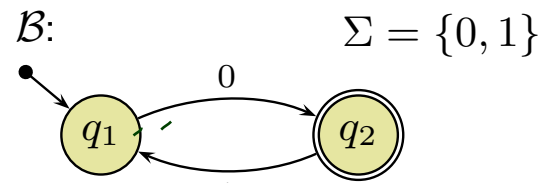
Excursion: Büchi Automata

From Finite Automata to Symbolic Büchi Automata



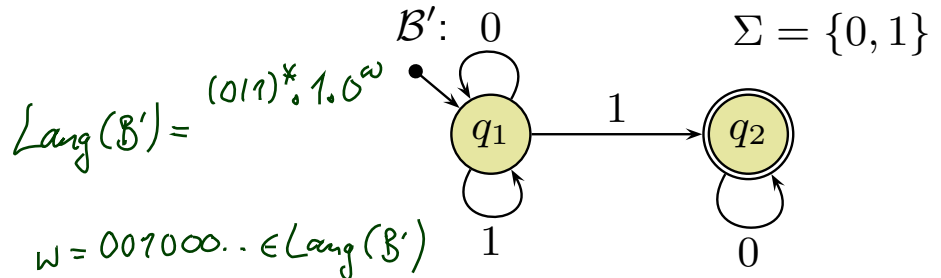
$\text{Lang}(A) = 0.(1.0)^*$
 $1 \notin \text{Lang}(A)$

Büchi
 infinite words



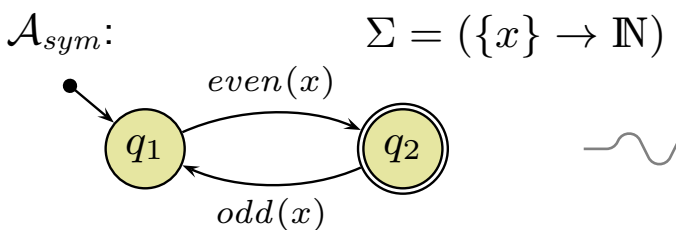
$w = 01010101\dots \in \text{Lang}(B)$
 $\text{Lang}(B) = 0.(1.0)^\omega$ ← infinite sequence

symbolic



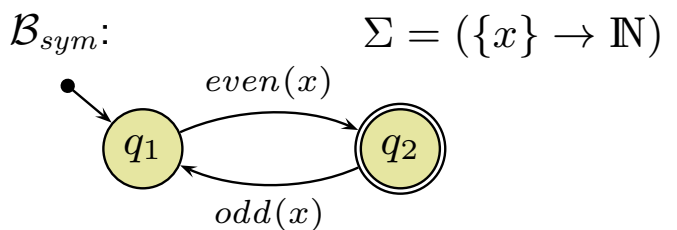
$\text{Lang}(B') = (011)^*.1.0^\omega$
 $w = 001000\dots \in \text{Lang}(B')$

symbolic



$w = \underbrace{(x \mapsto 0)}_{\in \Sigma} (x \mapsto 2) (x \mapsto 2)$

Büchi
 infinite words



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

$$\mathcal{B} = (\text{Expr}_{\mathcal{B}}(X), X, \underbrace{Q, q_{ini}, \rightarrow, Q_F})$$

where

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

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 $\beta : X \rightarrow \mathcal{D}(X)$ of logical variables,

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

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

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

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

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

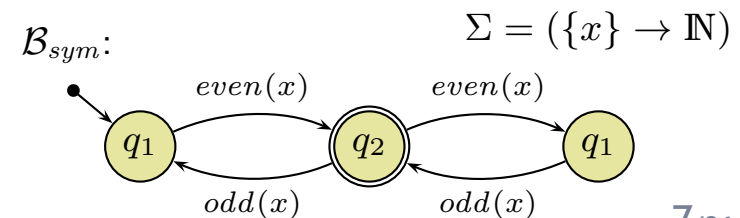
is called **run of \mathcal{B} over w** under valuation $\beta : X \rightarrow \mathcal{D}(X)$ if and only if

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

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

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

Example:



The Language of a TBA

Definition.

We say TBA $\mathcal{B} = (\text{Expr}_{\mathcal{B}}(X), X, Q, q_{ini}, \rightarrow, Q_F)$ **accepts** the word

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

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

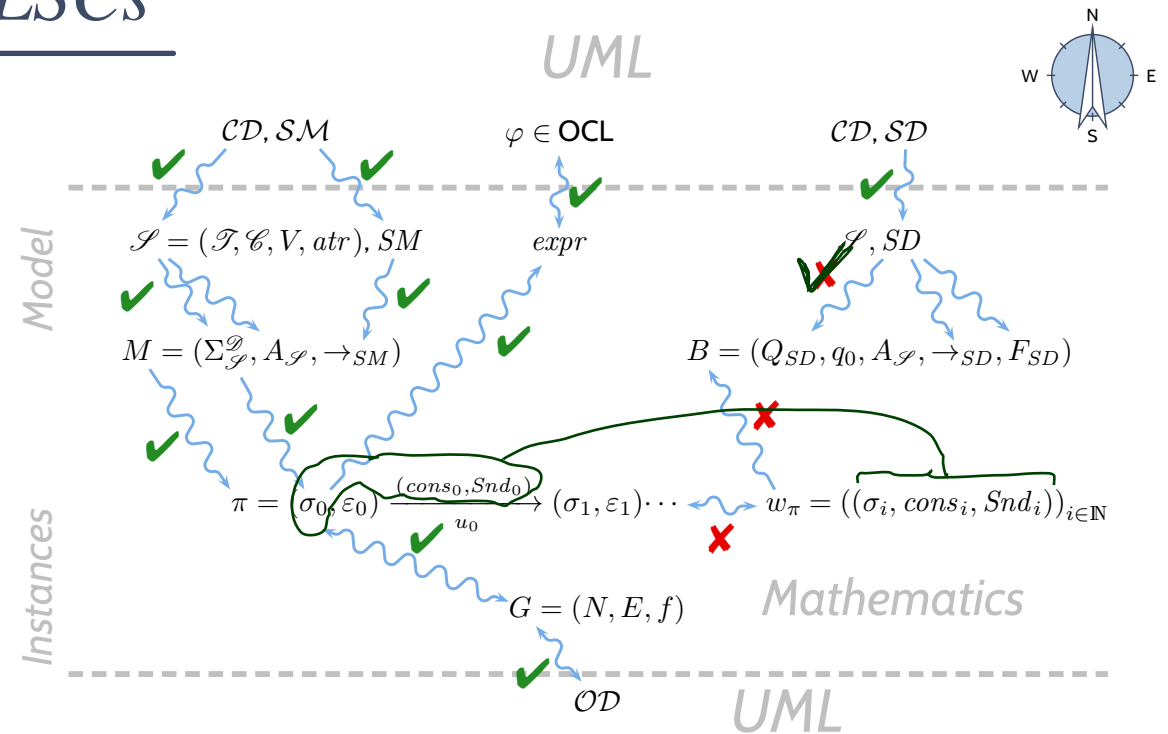
$$\varrho = (q_i)_{i \in \mathbb{N}_0}$$

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 $\mathcal{L}(\mathcal{B}) \subseteq (\text{Expr}_{\mathcal{B}} \rightarrow \mathbb{B})^\omega$ of words that are accepted by \mathcal{B} the **language of \mathcal{B}** .

TBA-based Semantics of LSCs



Plan:

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

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

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

(iii) define language $\mathcal{L}(\mathcal{L})$ of \mathcal{L} in terms of $\mathcal{L}(\mathcal{B}_{\mathcal{L}})$,

in particular taking activation condition and activation mode into account.

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

• Then $\mathcal{M} \models \mathcal{L}$ (**universal**) if and only if $\mathcal{L}(\mathcal{M}) \subseteq \mathcal{L}(\mathcal{L})$.

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

Language of UML Model

The Language of a Model

Recall: A UML model $\mathcal{M} = (\mathcal{C}\mathcal{D}, \mathcal{I}\mathcal{M}, \mathcal{O}\mathcal{D})$ and a structure \mathcal{D} denote a set $[[\mathcal{M}]]$ of (initial and consecutive) **computations** of the form

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

$$a_i = (\text{cons}_i, \text{Snd}_i, u_i) \in \underbrace{2^{\mathcal{D}(\mathcal{E})} \times 2^{(\mathcal{D}(\mathcal{E}) \cup \{*,+\}) \times \mathcal{D}(\mathcal{C})} \times \mathcal{D}(\mathcal{C})}_{=: \tilde{A}}.$$

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

Definition. Let $\mathcal{M} = (\mathcal{C}\mathcal{D}, \mathcal{I}\mathcal{M}, \mathcal{O}\mathcal{D})$ be a UML model and \mathcal{D} a structure. Then

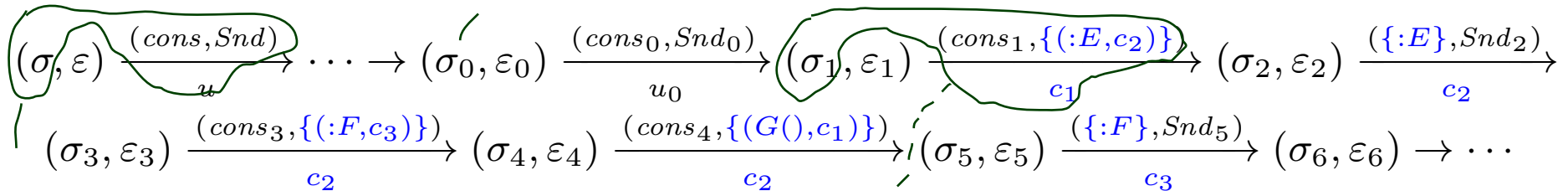
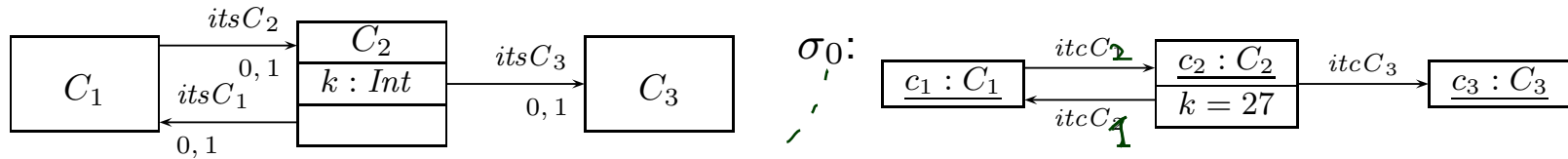
$$\mathcal{L}(\mathcal{M}) := \{(\sigma_i, u_i, \text{cons}_i, \text{Snd}_i)_{i \in \mathbb{N}_0} \in (\Sigma_{\mathcal{D}}^{\mathcal{D}} \times \tilde{A})^\omega \mid \\ \exists (\varepsilon_i)_{i \in \mathbb{N}_0} : (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(\text{cons}_0, \text{Snd}_0)} (\sigma_1, \varepsilon_1) \cdots \in [[\mathcal{M}]]\}$$

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

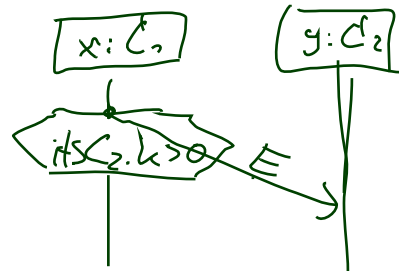
Example: Language of a Model

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

CD:



$(\sigma, u, cons, Snd), \dots, (\sigma_0, u_0, cons_0, Snd_0), (\sigma_1, c_1, cons_1, \{(:E, c_2)\}), \dots \in \text{Lang}(\mathcal{M})$



$\beta = \{x \mapsto c_1, y \mapsto c_2\}$
 $E_{x,y}^\dagger \wedge (x.itsC_2.k > 0)$
 $a \models_\beta$

Words over Signature

Definition. Let $\mathcal{S} = (\mathcal{I}, \mathcal{C}, V, atr, \mathcal{E})$ be a signature and \mathcal{D} a structure of \mathcal{S} .

A **word** over \mathcal{S} and \mathcal{D} is an infinite sequence

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

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

Satisfaction of Signal and Attribute Expressions

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

Then

- $(\sigma, u, cons, Snd) \models_{\beta} \text{true}$
- $(\sigma, u, cons, Snd) \models_{\beta} \psi$ 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 $(\sigma, u, cons, Snd) \models_{\beta} \psi_2$
- $(\sigma, \underline{u}, cons, \underline{Snd}) \models_{\beta} E_{x,y}^!$ if and only if $\beta(x) = u \wedge \exists e \in \mathcal{D}(E) \bullet (e, \beta(y)) \in Snd$

E-identity
- $(\sigma, \underline{u}, \underline{cons}, \underline{Snd}) \models_{\beta} E_{x,y}^?$ if and only if $\beta(y) = u \wedge \underline{cons} \subset \mathcal{D}(E) \wedge \underline{cons} \neq \emptyset$

"cons is an E-identity"

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.

Definition. A TBA

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

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

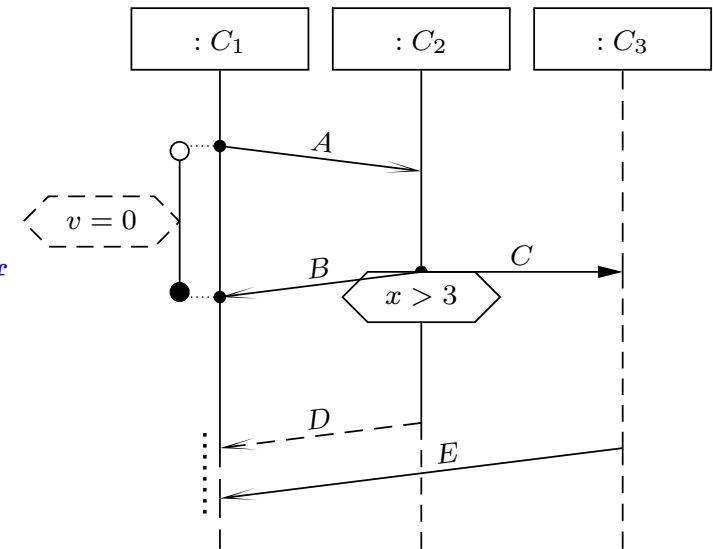
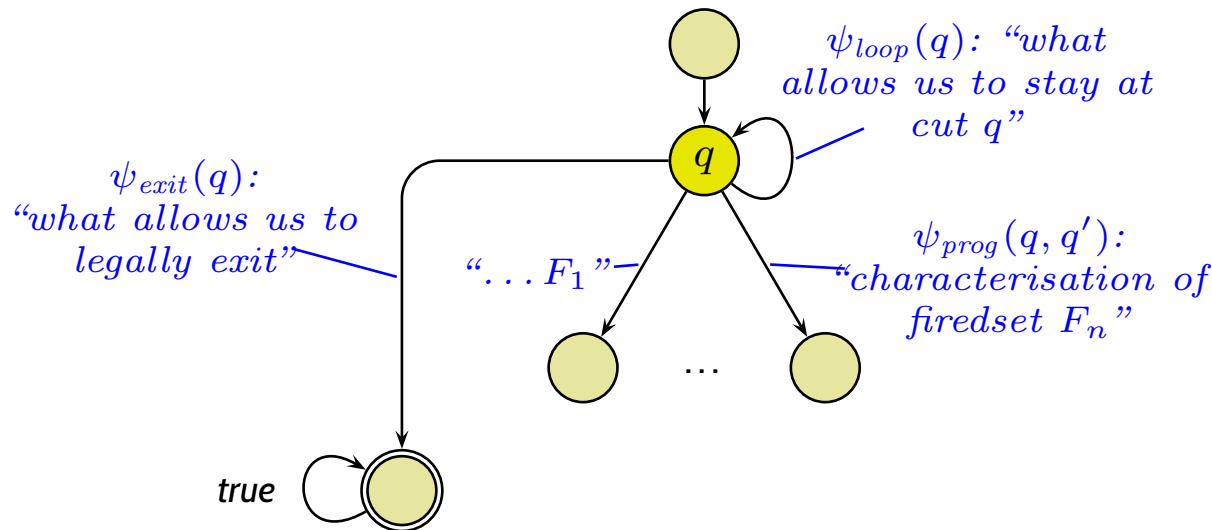
TBA Construction Principle

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

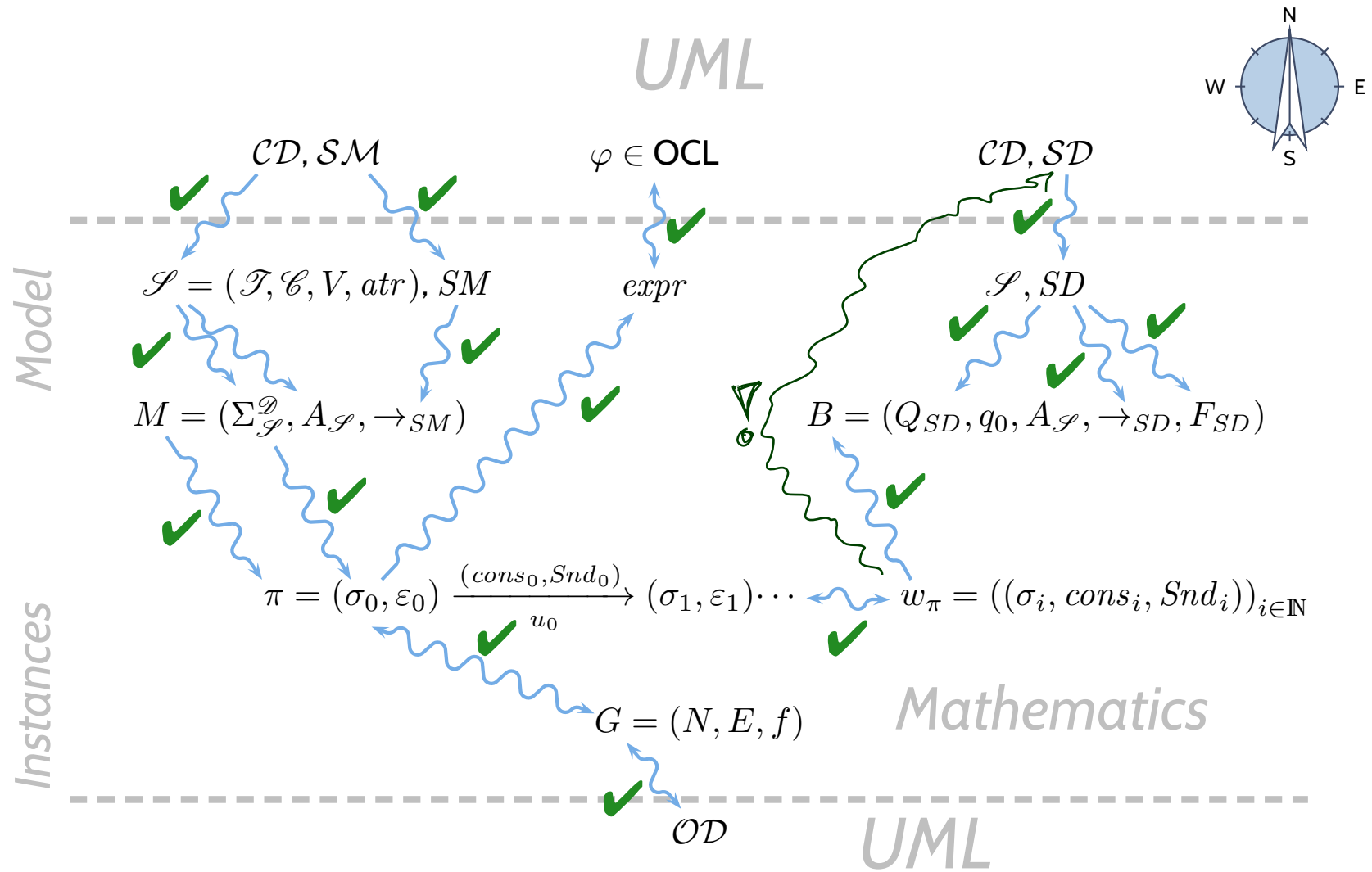
- Q is the **set of cuts** of \mathcal{L} , q_{ini} is the **instance heads** cut,
- $Expr_{\mathcal{B}} = \mathcal{E}!?(X)$, *signal/attribute expressions*
- \rightarrow consists of **loops**, **progress transitions** (from \rightsquigarrow_F), and **legal exits** (cold cond./local inv.),
- $F = \{C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

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

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



Course Map



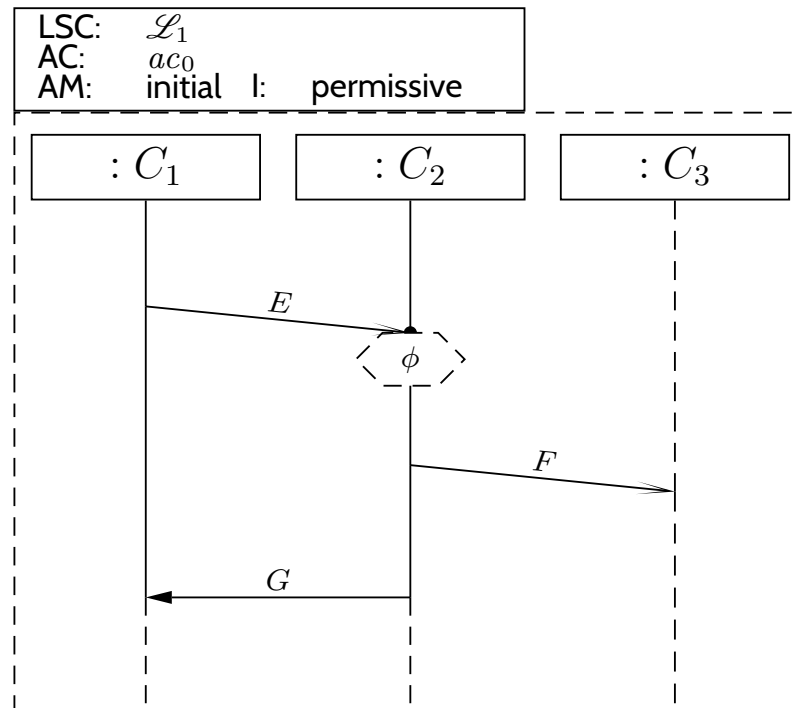
Live Sequence Charts — Full LSC Semantics

Full LSCs

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

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

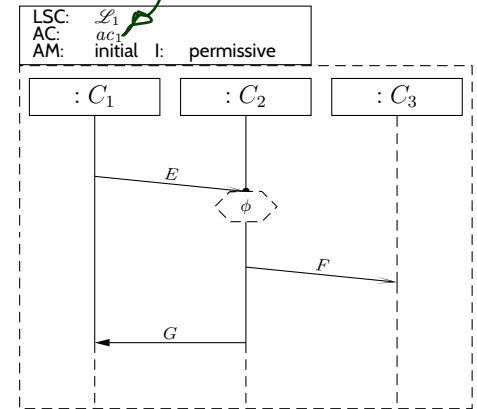
Concrete syntax:



Full LSCs

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

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

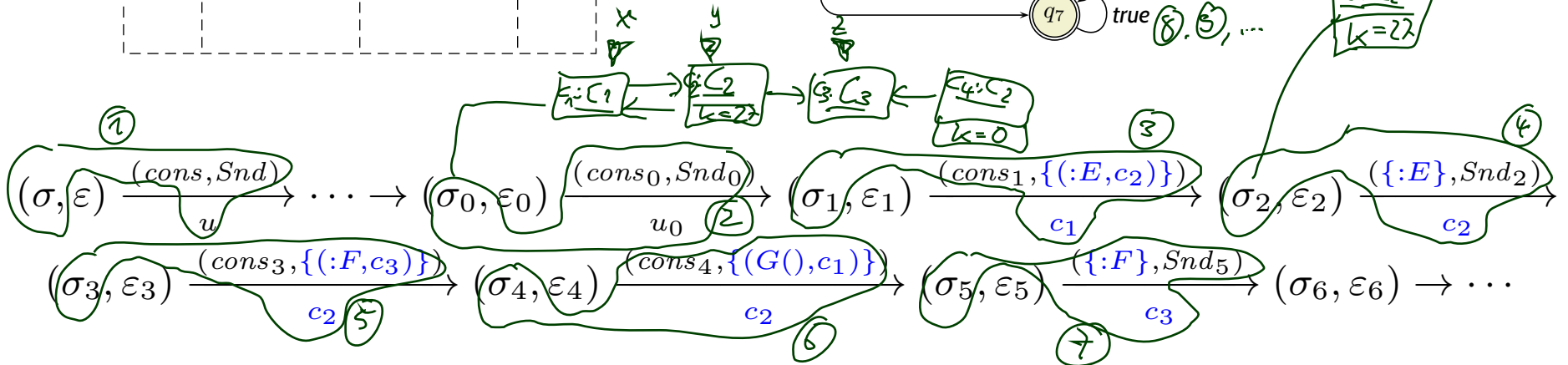
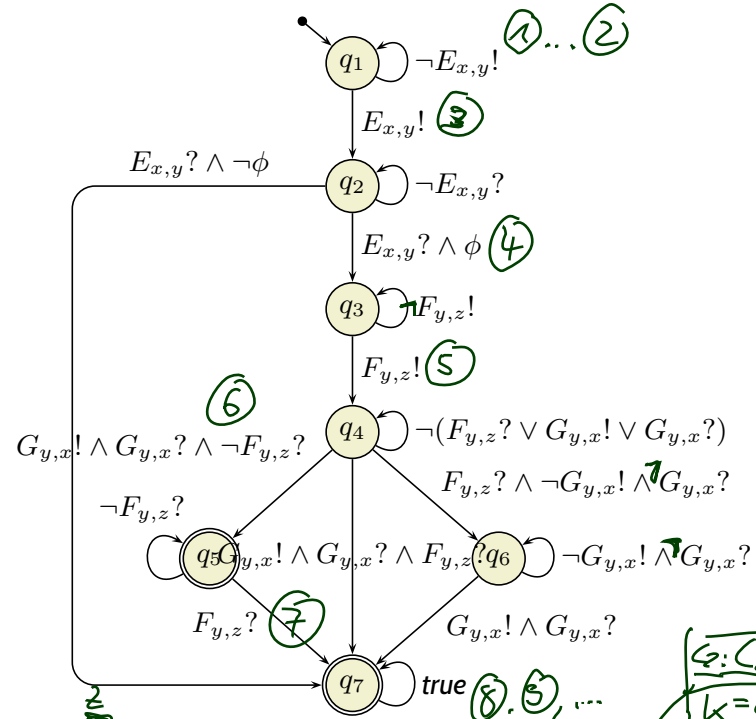
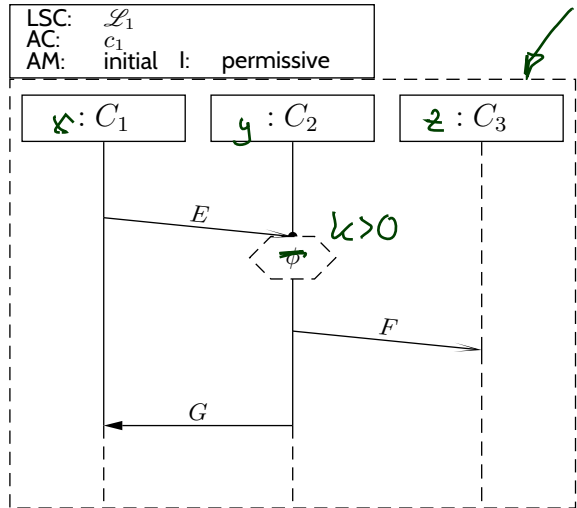


A **set of words** $W \subseteq (\text{Expr}_{\mathcal{B}} \rightarrow \mathbb{B})^\omega$ is **accepted** by \mathcal{L} if and only if

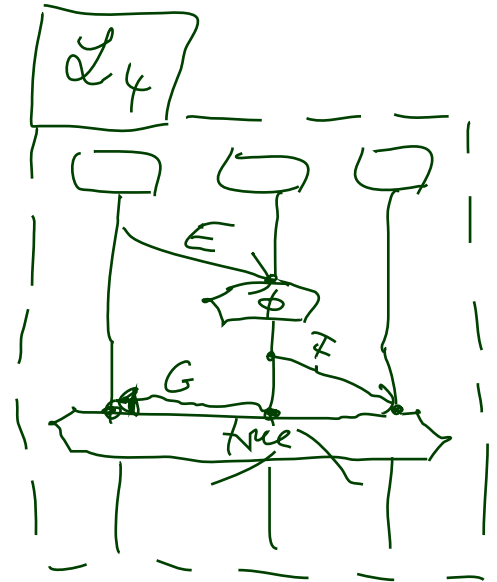
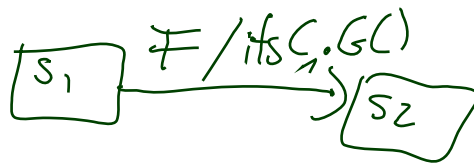
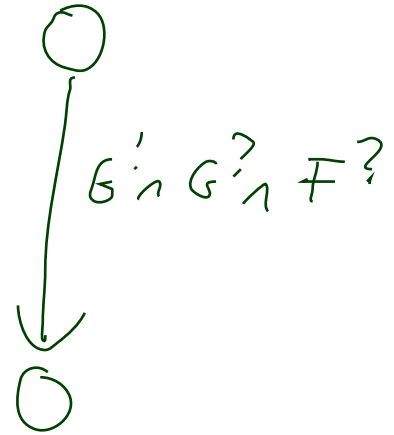
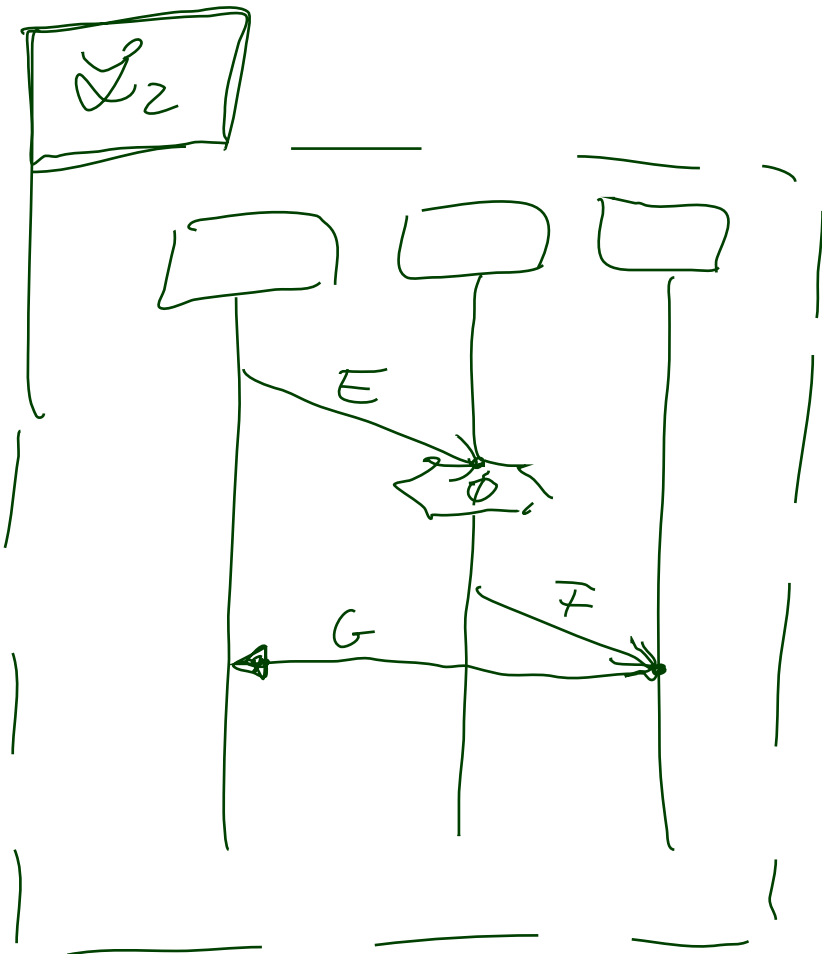
	$am = \text{initial}$	$am = \text{invariant}$
cold $\Theta_{\mathcal{L}}$	$\exists \beta \exists w \in W \bullet w^0 \models_{\beta} ac \wedge \neg \psi_{\text{exit}}(C_0) \wedge w^0 \models_{\beta} \psi_{\text{prog}}(\emptyset, C_0) \wedge w/1 \in \mathcal{L}(\mathcal{B}(\mathcal{L}))$	$\exists \beta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

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

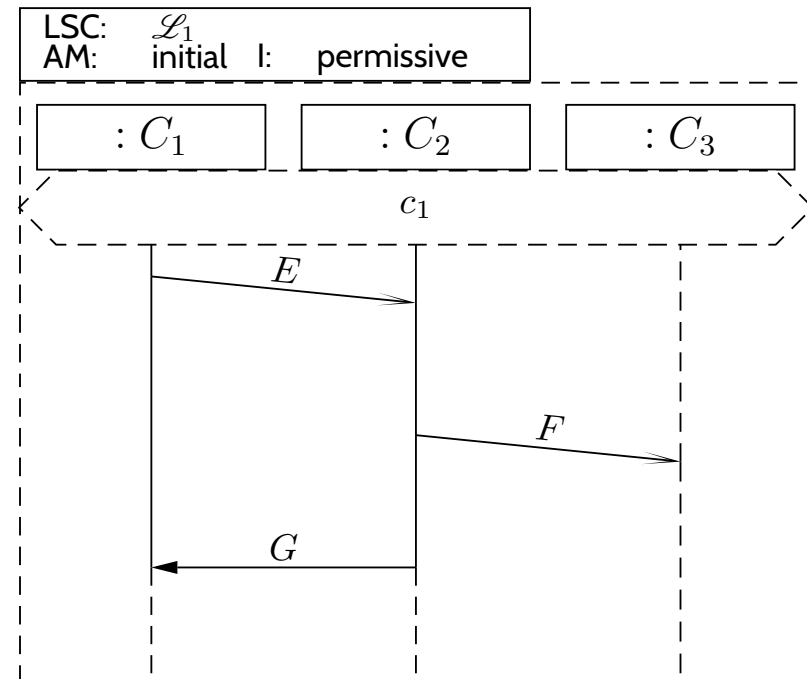
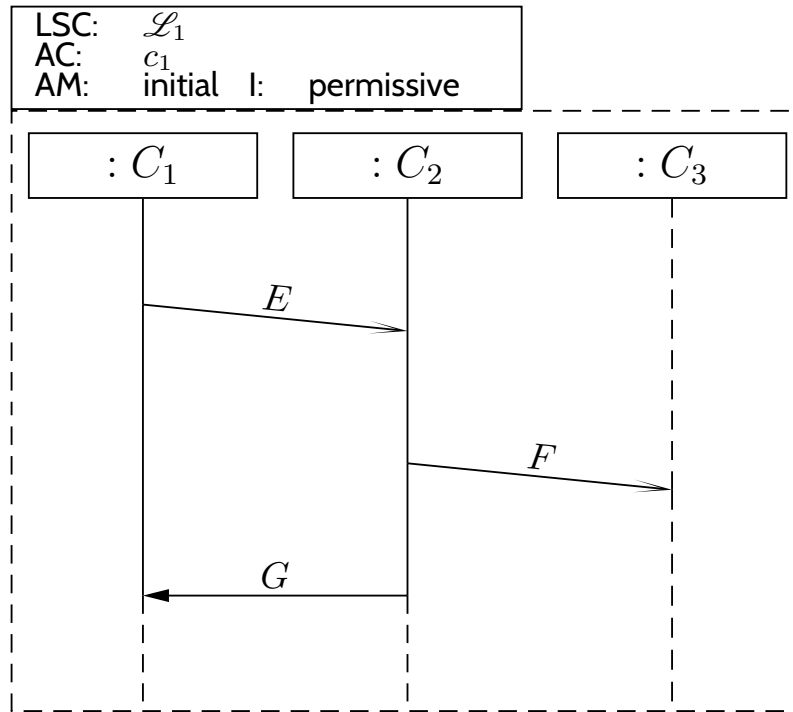
Full LSC Semantics: Example



$x \mapsto c_1$
 $y \mapsto c_2$
 $z \mapsto c_3$

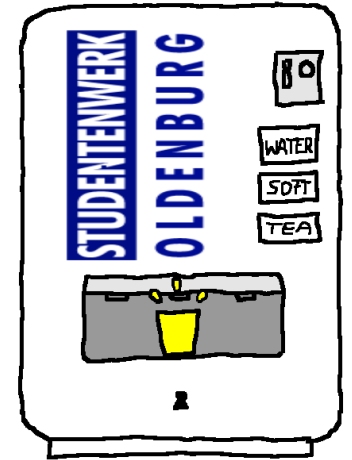
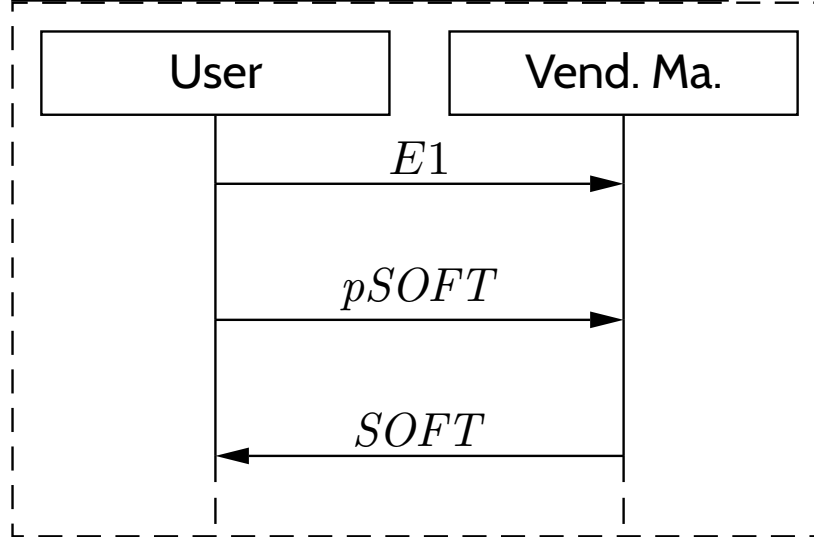


Note: Activation Condition



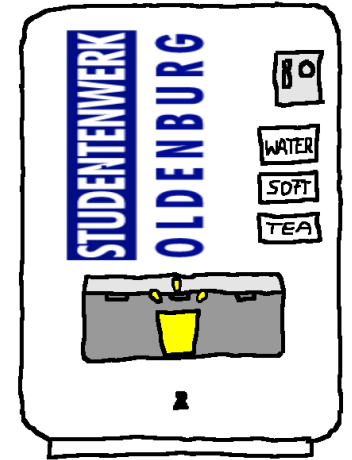
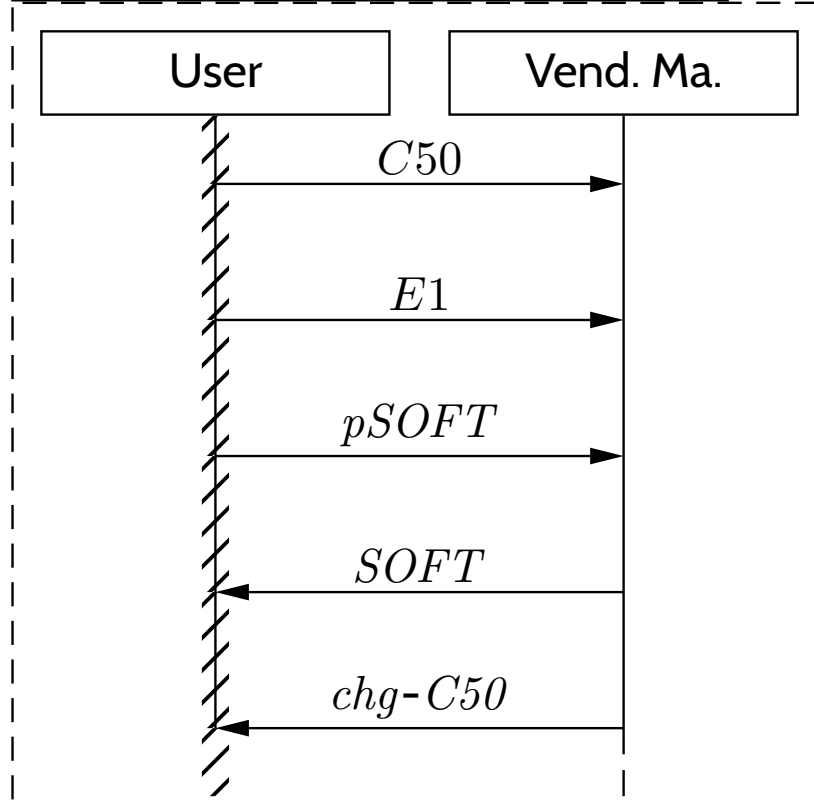
Existential LSC Example: Buy A Softdrink

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

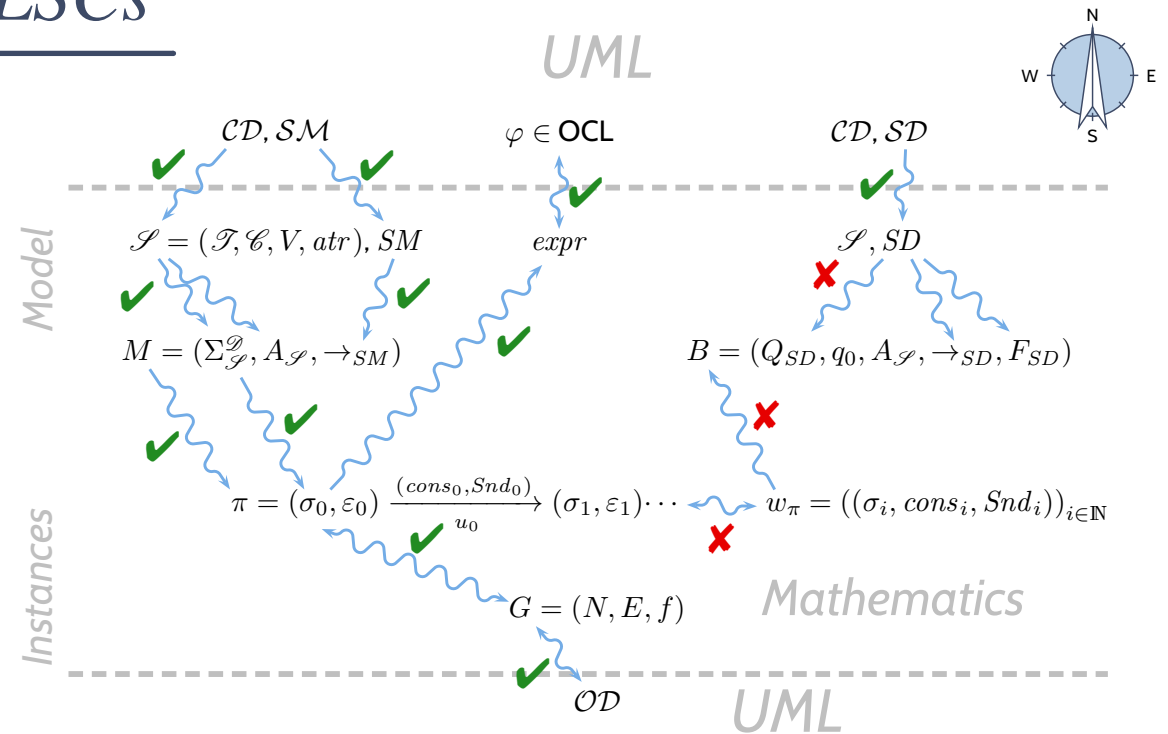


Existential LSC Example: Get Change

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



TBA-based Semantics of LSCs



Plan:

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

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

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

(iii) define language $\mathcal{L}(\mathcal{L})$ of \mathcal{L} **in terms of** $\mathcal{L}(\mathcal{B}_{\mathcal{L}})$,

in particular taking activation condition and activation mode into account.

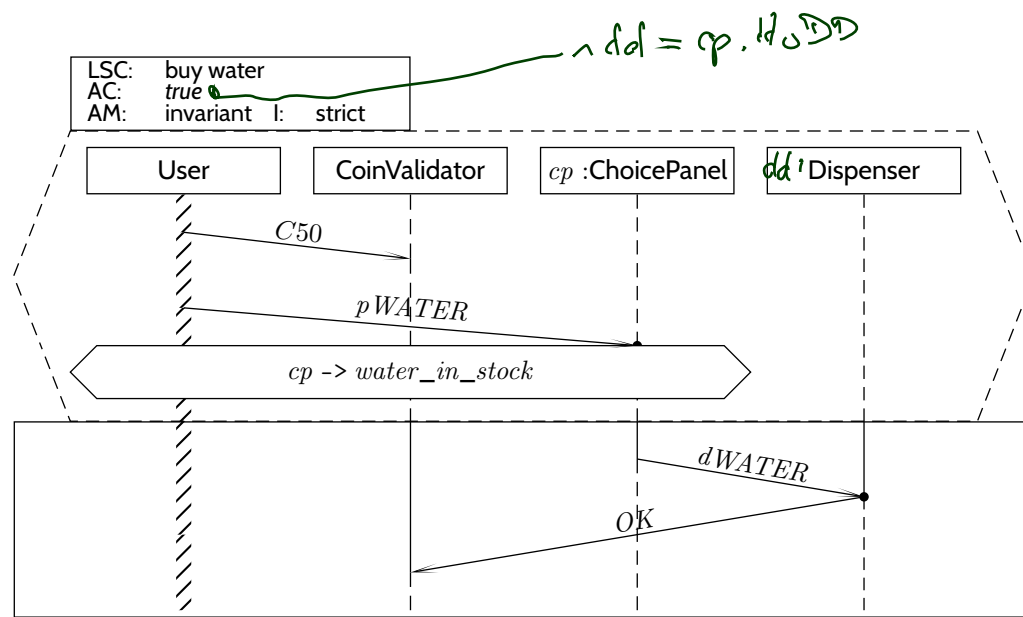
(iv) define language $\mathcal{L}(\mathcal{M})$ of a UML model.

• Then $\mathcal{M} \models \mathcal{L}$ (**universal**) if and only if $\mathcal{L}(\mathcal{M}) \subseteq \mathcal{L}(\mathcal{L})$.

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

Live Sequence Charts — Precharts

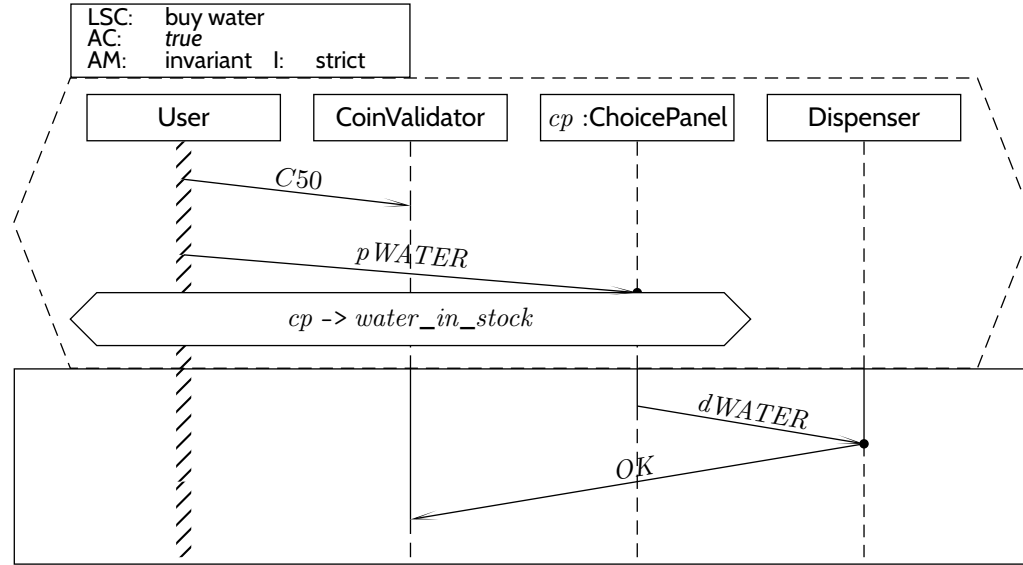
Pre-Charts



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

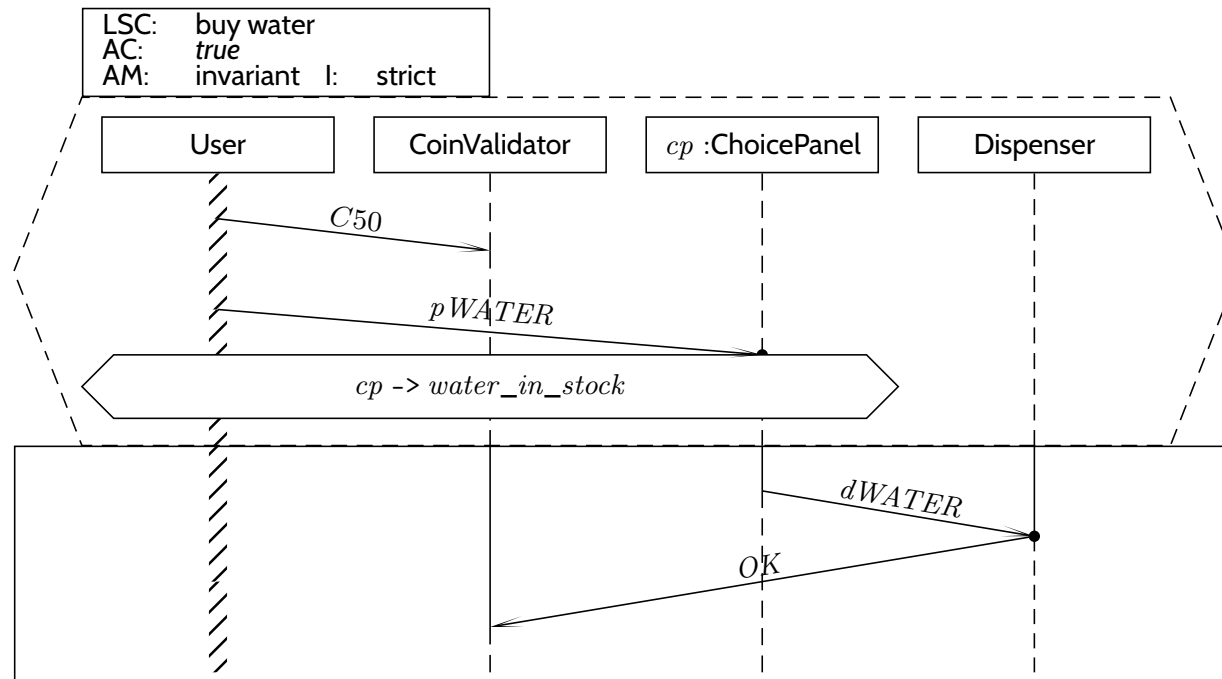
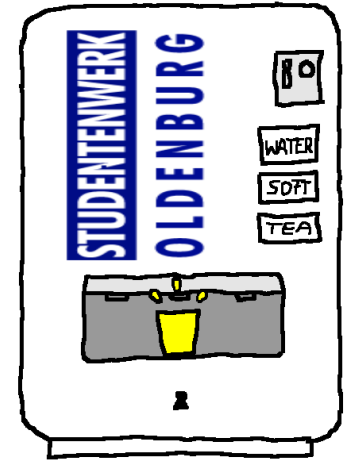
- **pre-chart** $PC = ((L_P, \preceq_P, \sim_P), \mathcal{I}_P, \mathcal{S}, \text{Msg}_P, \text{Cond}_P, \text{LocInv}_P, \Theta_P)$ (possibly empty),
- **main-chart** $MC = ((L_M, \preceq_M, \sim_M), \mathcal{I}_M, \mathcal{S}, \text{Msg}_M, \text{Cond}_M, \text{LocInv}_M, \Theta_M)$ (non-empty),
-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

Pre-Charts Semantics

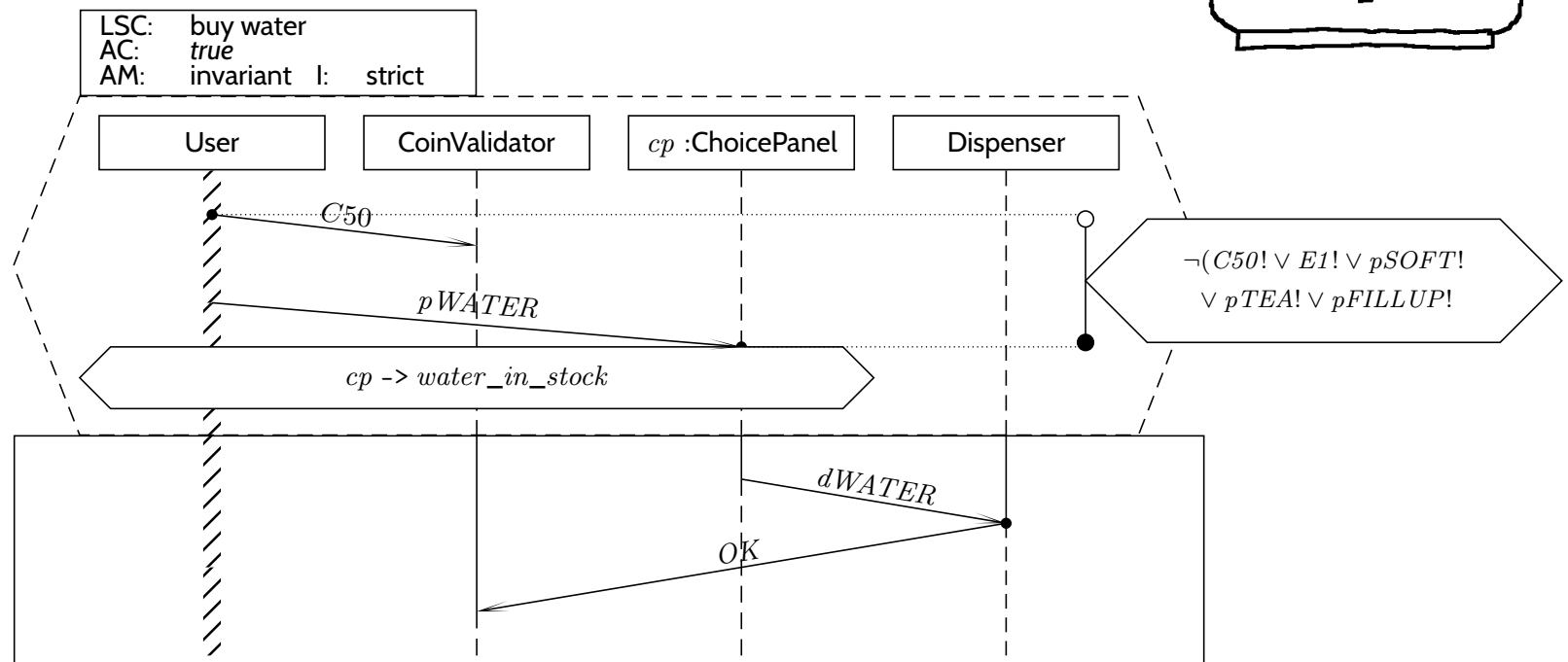
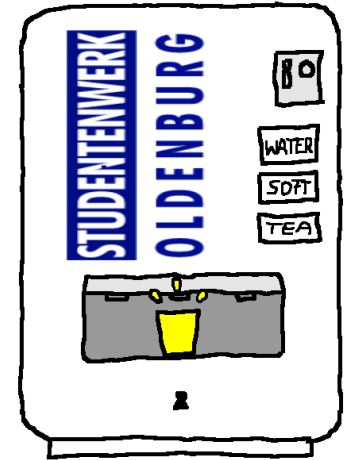


	$am = \text{initial}$	$am = \text{invariant}$
$\ominus_{\mathcal{L}} = \text{cold}$	$\exists w \in W \exists m \in \mathbb{N}_0 \bullet$ $\wedge w^0 \models ac \wedge \neg \psi_{exit}(C_0^P) \wedge \psi_{prog}(\emptyset, C_0^P)$ $\wedge w^1, \dots, w^m \in \mathcal{L}(\mathcal{B}(PC))$ $\wedge w^{m+1}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

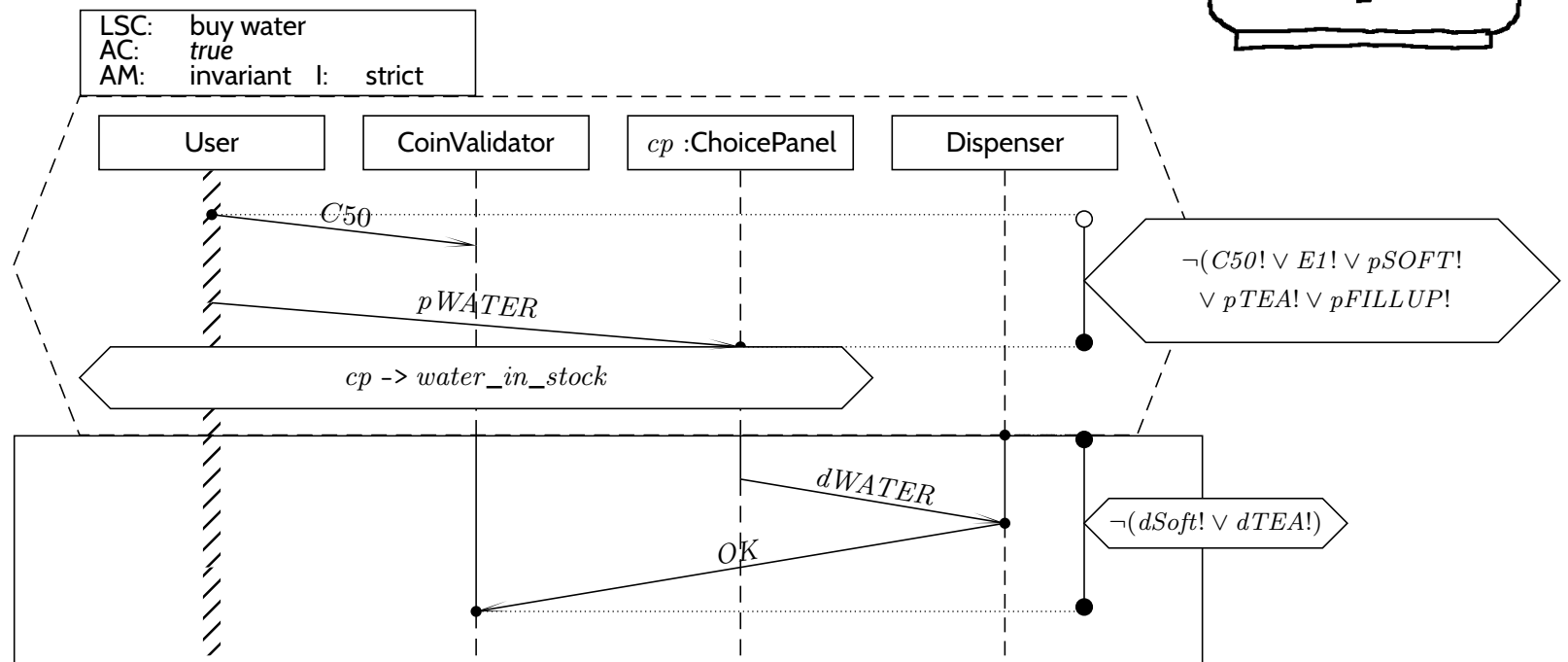
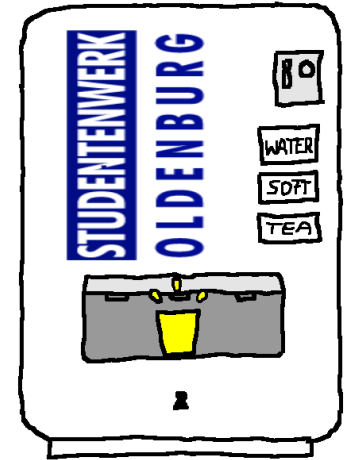
Universal LSC: Example



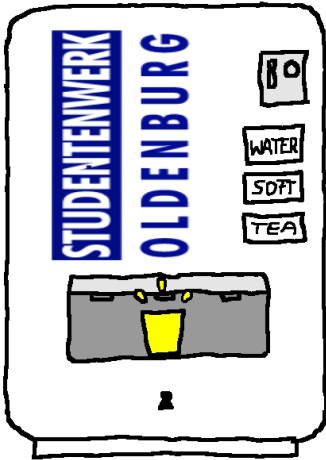
Universal LSC: Example



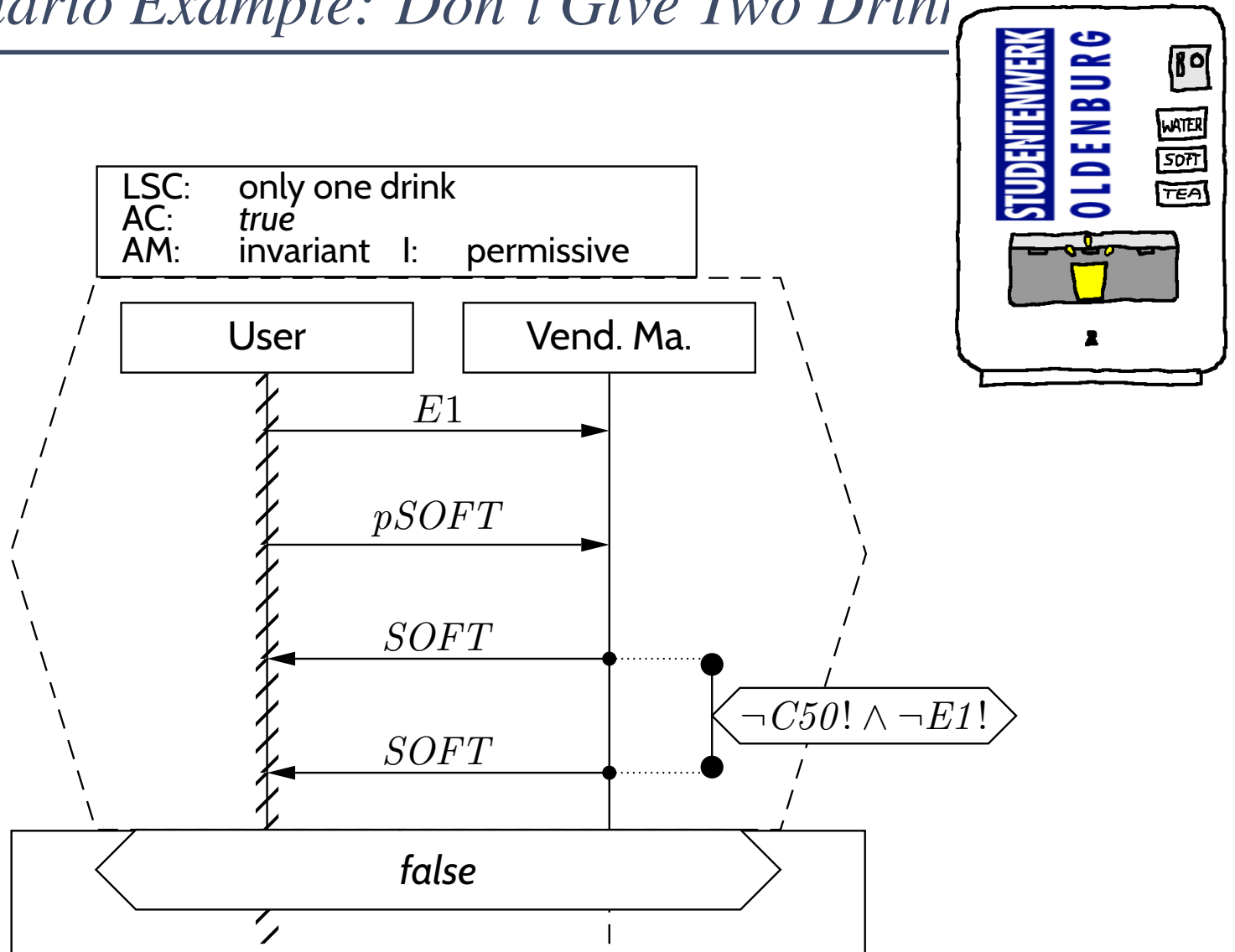
Universal LSC: Example



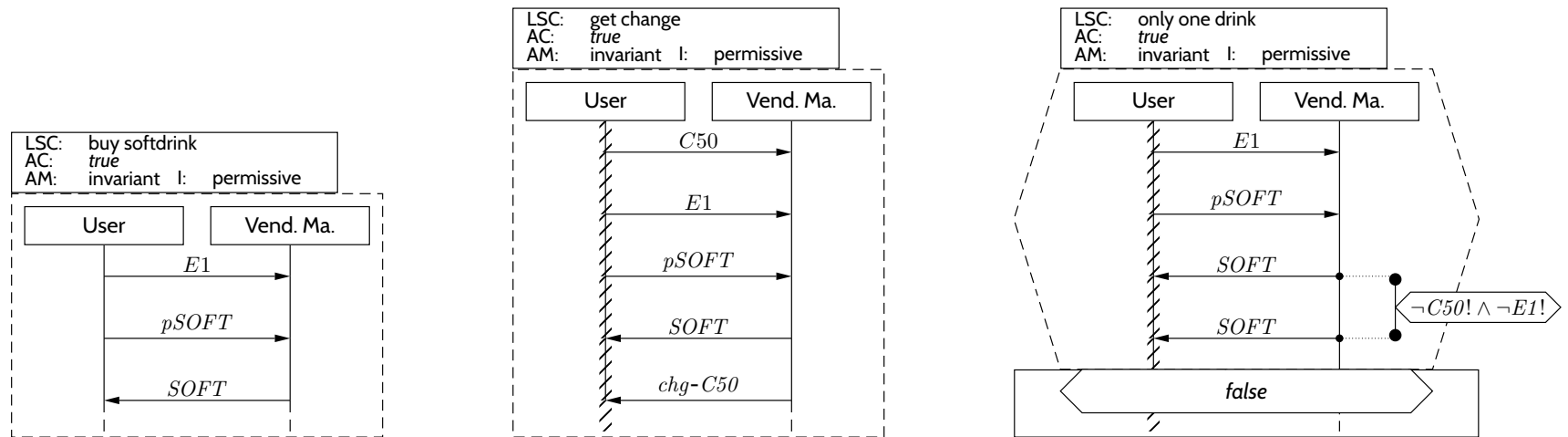
Forbidden Scenario Example: Don't Give Two Drinks



Forbidden Scenario Example: Don't Give Two Drinks

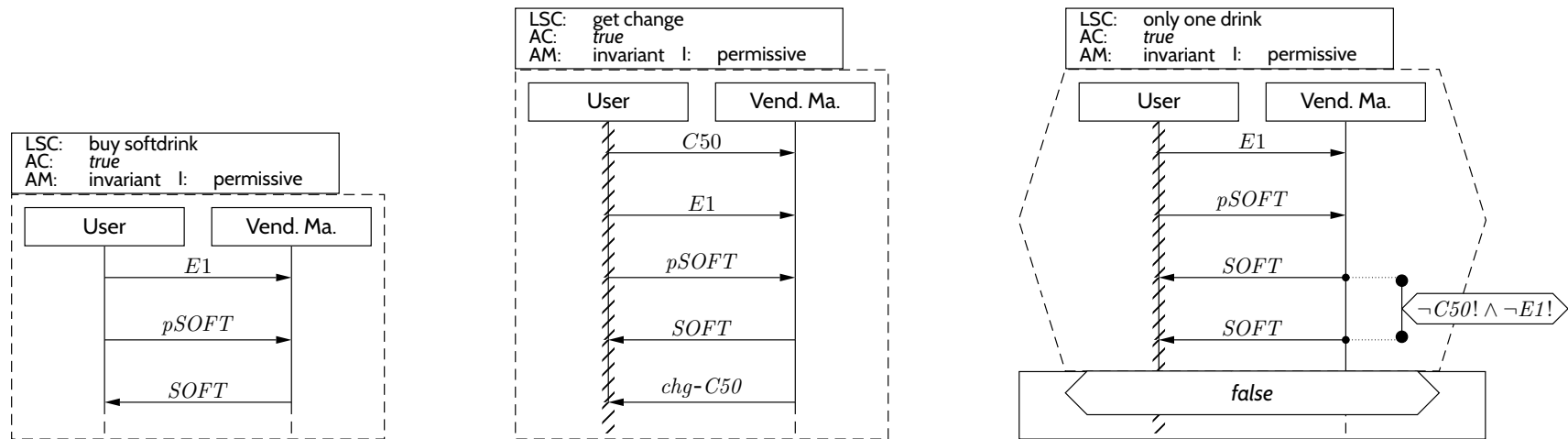


Note: Sequence Diagrams and (Acceptance) Test



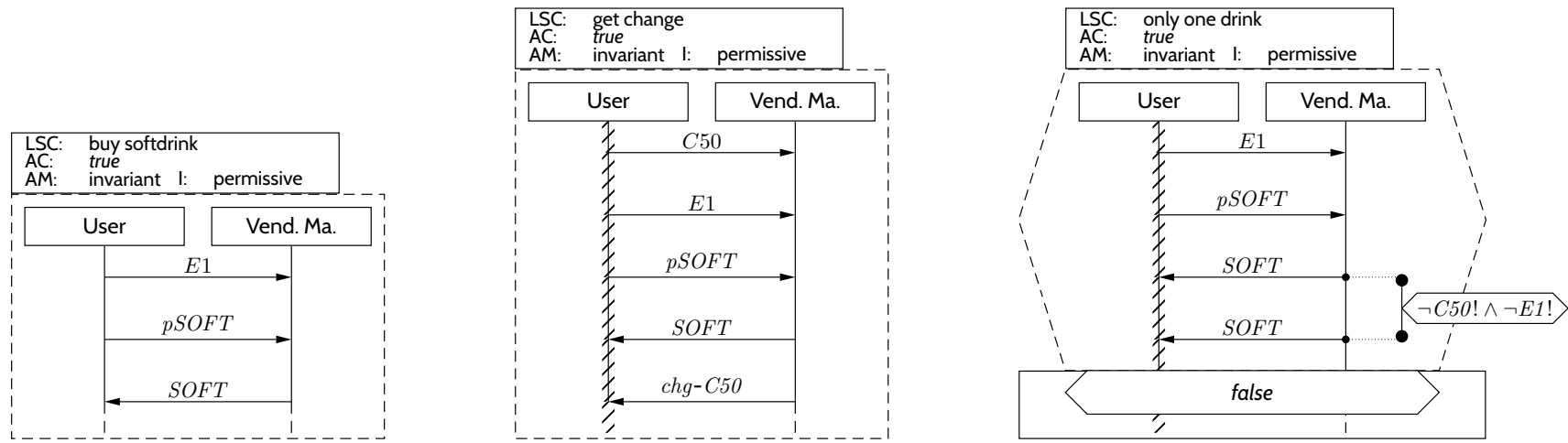
- **Existential LSCs*** may hint at **test-cases** for the **acceptance test!**
 (*: as well as (positive) scenarios in general, like use-cases)

Note: Sequence Diagrams and (Acceptance) Test



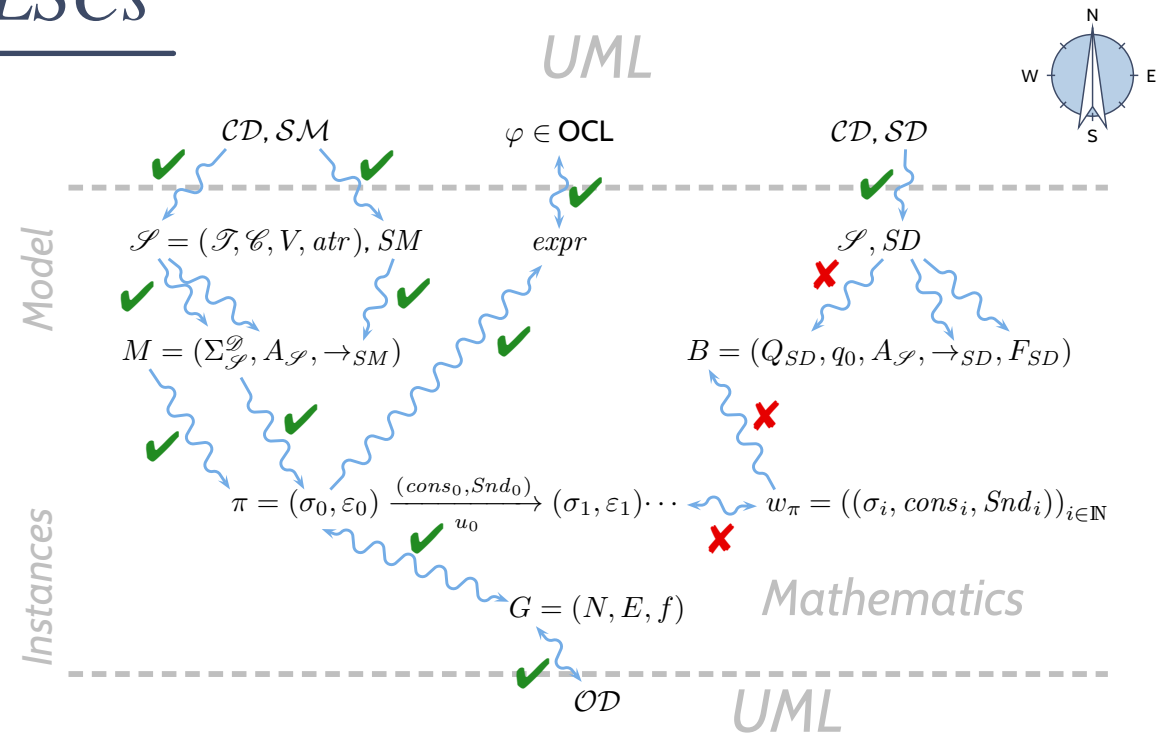
- **Existential** LSCs* may hint at **test-cases** for the **acceptance test!**
 (*: as well as (positive) scenarios in general, like use-cases)
- **Universal** LSCs (and negative/anti-scenarios) in general need **exhaustive analysis!**

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 \mathcal{L} with body

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

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

(iii) define language $\mathcal{L}(\mathcal{L})$ of \mathcal{L} in terms of $\mathcal{L}(\mathcal{B}_{\mathcal{L}})$,

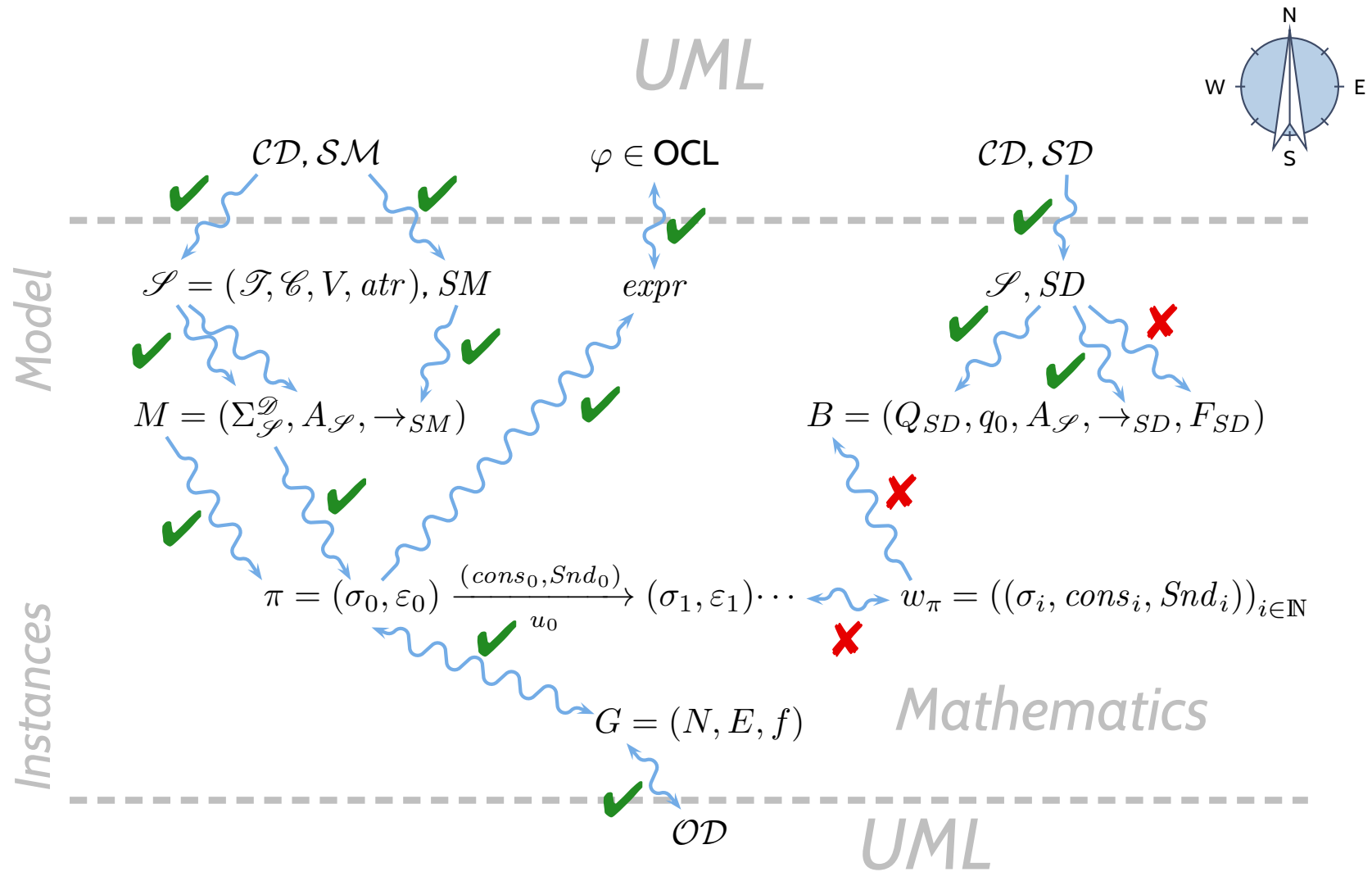
in particular taking activation condition and activation mode into account.

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

• Then $\mathcal{M} \models \mathcal{L}$ (**universal**) if and only if $\mathcal{L}(\mathcal{M}) \subseteq \mathcal{L}(\mathcal{L})$.

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

Course Map



Tell Them What You've Told Them. . .

- 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.