Software Design, Modelling and Analysis in UML

Lecture 19: Live Sequence Charts II

2014-01-29

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

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

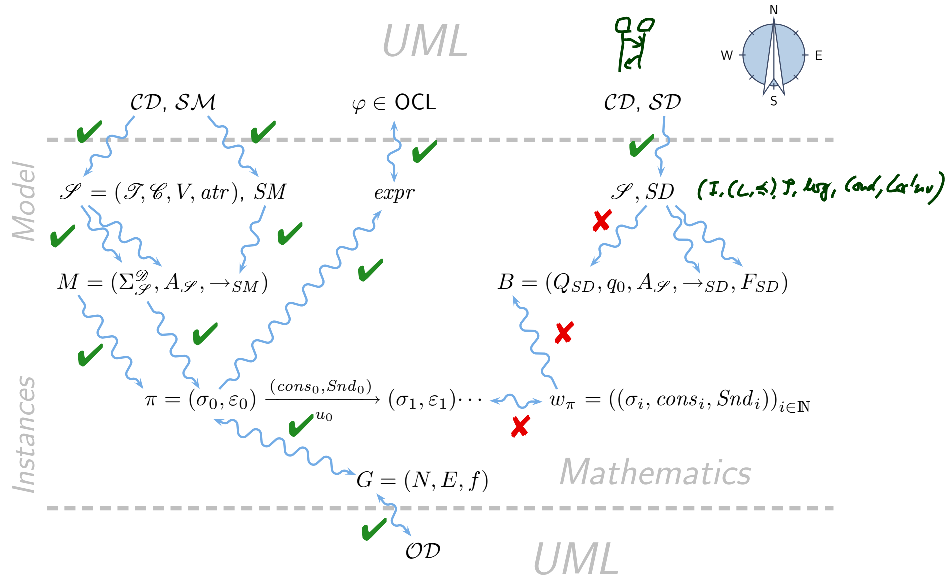
Last Lecture:

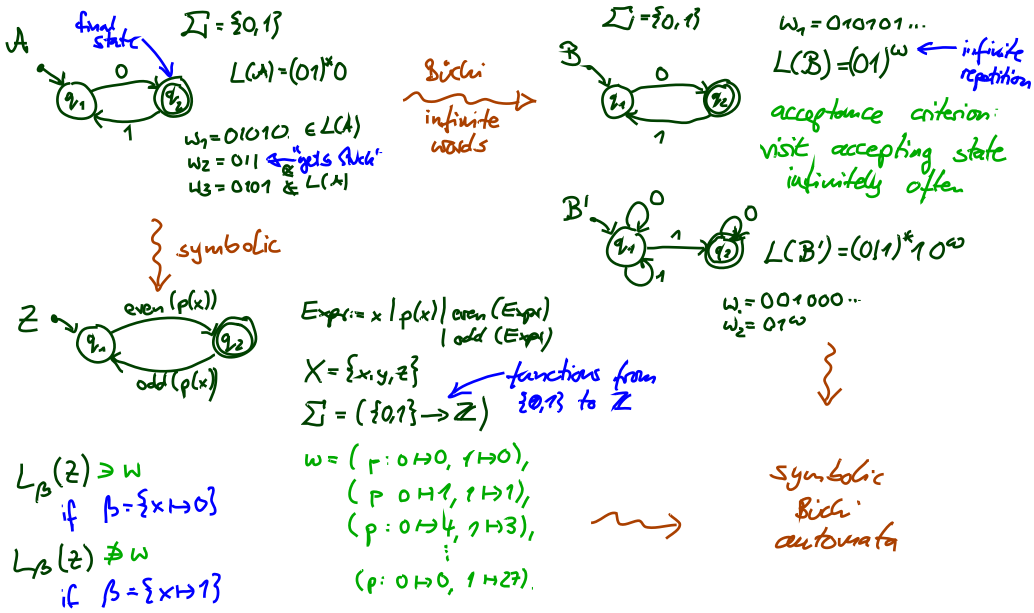
- LSC intuition
- LSC abstract syntax

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - What does this LSC mean?
  - Are this UML model's state machines consistent with the interactions?
  - Please provide a UML model which is consistent with this LSC.
  - What is: activation, hot/cold condition, pre-chart, etc.?
- **Content:**
  - Symbolic Büchi Automata (TBA) and its (accepted) language.
  - Words of a model.
  - LSC formal semantics.

Course Map


UML
CD, SM
φ ∈ OCL
CD, SD
N
W
E
S
Model
𝒮 = (𝒯, 𝒞, V, atr), SM
expr
𝒮, SD
(I, (L, ≤), J, Msg, Cond, LocInv)
M = (Σ^𝒟_𝒮, A_𝒮, →_SM)
B = (Q_SD, q_0, A_𝒮, →_SD, F_SD)
Instances
π = (σ_0, ε_0) —(cons_0, Snd_0) / u_0→ (σ_1, ε_1) ···
w_π = ((σ_i, cons_i, Snd_i))_{i∈ℕ}
G = (N, E, f)
Mathematics
OD
UML


A
final state
Σ = {0,1}
L(A) = (01)*0
w_1 = 01010 ∈ L(A)
w_2 = 011 ∉ L(A) "gets stuck"
w_3 = 0101 ∉ L(A)
Büchi
infinite words
B
Σ = {0,1}
w_1 = 010101...
L(B) = (01)^ω ← infinite repetition
acceptance criterion:
visit accepting state infinitely often
B'
L(B') = (0|1)* 1 0^ω
w_1 = 001000...
w_2 = 01^ω
symbolic
Z
even(p(x))
odd(p(x))
Expr ::= x | p(x) | even(Expr) | odd(Expr)
X = {x, y, z}
Σ = ({0,1} → Z) ← functions from {0,1} to Z
w = (p: 0↦0, 1↦0),
(p: 0↦1, 1↦1),
(p: 0↦4, 1↦3),
⋮
(p: 0↦0, 1↦27).
L_β(Z) ∋ w
if β = {x↦0}
L_β(Z) ∌ w
if β = {x↦1}
symbolic Büchi automata

Excursus: Symbolic Büchi Automata (over Signature)

– 19 – 2014-01-29 – main –

Symbolic Büchi Automata

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

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

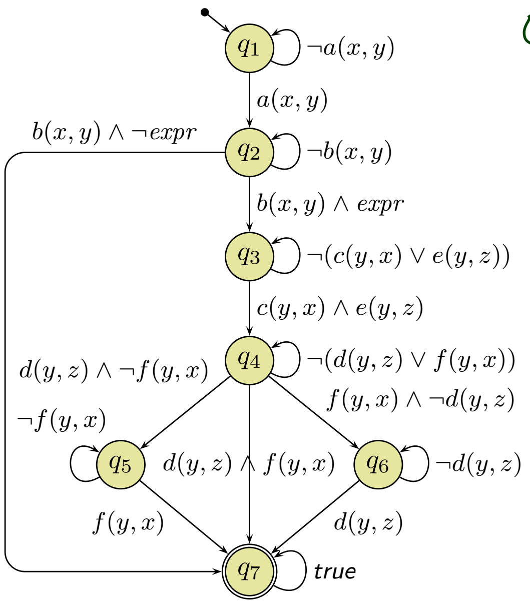
where

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

– 19 – 2014-01-29 – Stba –

TBA Example

$(Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \rightarrow, Q_F)$, $(q, \psi, q') \in \rightarrow$,



$Q = \{q_1, \ldots, q_7\}$

$q_{ini} = q_1$

$Q_F = \{q_7\}$

$X = \{x, y, z\}$

$Expr(X)$: $a(x_1, x_2) \mid expr \mid \neg expr \mid \ldots$

$\rightarrow = \{ (q_1, \neg a(x,y), q_1), \ldots \}$

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

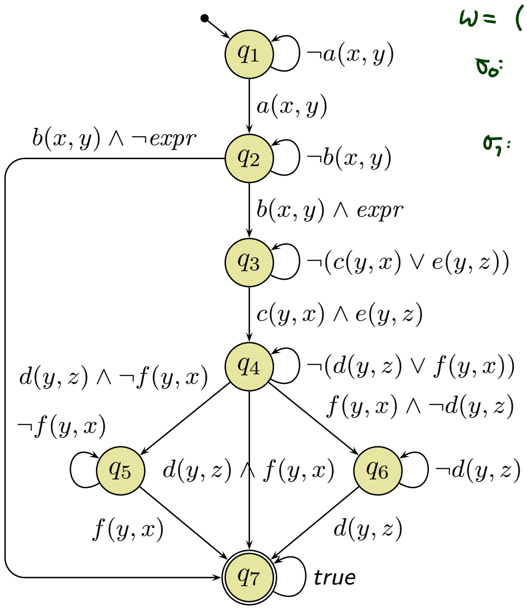
$$\text{either } \sigma \models_\beta expr \text{ or } \sigma \not\models_\beta expr.$$

An **infinite sequence**

$$w = (\sigma_i)_{i \in \mathbb{N}_0} \in \Sigma^\omega$$

over $(\Sigma, \cdot \models . \cdot)$ is called **word** for $Expr_{\mathcal{B}}(X)$.

Word Example



$w = (a: (1,2) \mapsto 0,\ (2,1) \mapsto 1, \dots$
$b: \dots$
$\vdots$
$f: \dots),$

σ_0:

σ_1:

Run of TBA over Word

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

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

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

An infinite sequence

states!

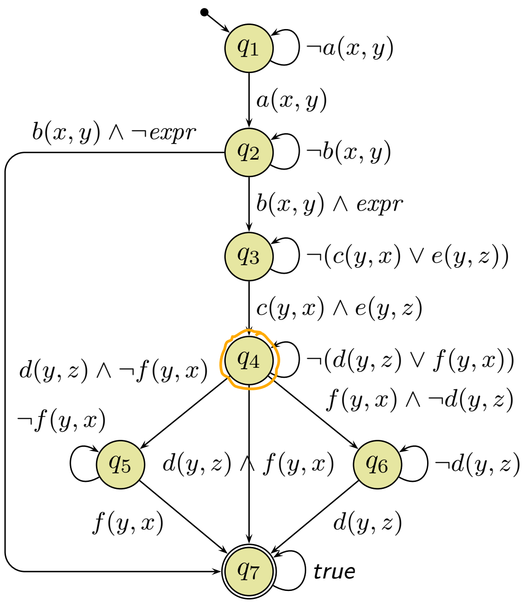
$$\varrho = q_0, q_1, q_2, \dots \in Q^\omega$$

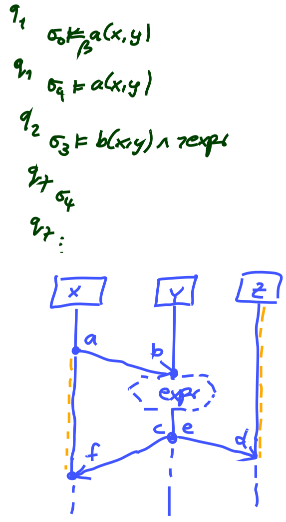
is called **run** of $\mathcal{B}$ over w under valuation $\beta : X \to \mathscr{D}(X)$ if and only if

- $q_0 = q_{ini}$,
- for each $i \in \mathbb{N}_0$ there is a transition $(q_i, \psi_i, q_{i+1}) \in \to$ of $\mathcal{B}$ such that $\sigma_i \models_\beta \psi_i$.

Run Example

$$\varrho = q_0, q_1, q_2, \ldots \in Q^\omega \text{ s.t. } \sigma_i \models_\beta \psi_i,\ i \in \mathbb{N}_0.$$





The Language of a TBA

Definition.
We say $\mathcal{B}$ **accepts** word w (under β) 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 $\mathcal{L}_\beta(\mathcal{B}) \subseteq \Sigma^\omega$ of words for $Expr_\mathcal{B}(X)$ that are accepted by $\mathcal{B}$ the **language of** $\mathcal{B}$.

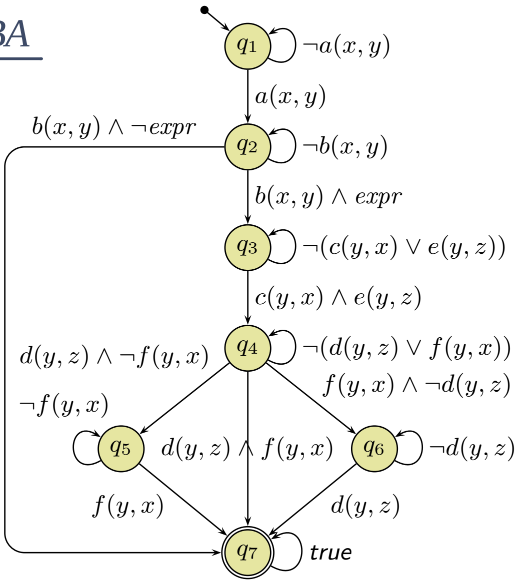
Language of the Example TBA

$\mathcal{L}_\beta(\mathcal{B})$ consists of the words

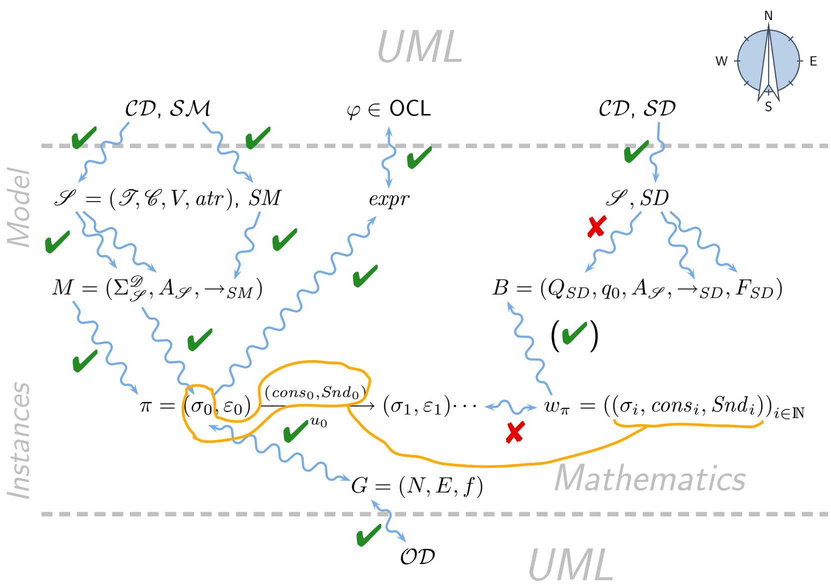
$$w = (\sigma_i)_{i \in \mathbb{N}_0}$$

where for $0 \leq n < m < k < \ell$ we have

- for $0 \leq i < n$, $\sigma_i \not\models_\beta E^!_{x,y}$
- $\sigma_n \models_\beta E^!_{x,y}$
- for $n < i < m$, $\sigma_i \not\models_\beta E^?_y$
- $\sigma_m \models_\beta E^?_y$
- for $m < i < k$, $\sigma_i \not\models_\beta F^!_{y,x}$
- $\sigma_k \models_\beta F^!_{y,x}$
- for $k < i < \ell$, $\sigma_i \not\models_\beta F^?_{x,y}$
- ...



Course Map



Back to Main Track: Language of a Model

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

$$(\sigma_i, cons_i, Snd_i)_{i \in \mathbb{N}_0}$$
$$\in \left(\Sigma_{\mathscr{S}}^{\mathscr{D}} \times 2^{\mathscr{D}(\mathscr{C}) \times Evs(\mathscr{E}, \mathscr{D}) \times \mathscr{D}(\mathscr{C})} \times 2^{\mathscr{D}(\mathscr{C}) \times Evs(\mathscr{E}, \mathscr{D}) \times \mathscr{D}(\mathscr{C})}\right)^{\omega}.$$

The Language of a Model

Recall: A UML model $\mathcal{M} = (\mathscr{CD}, \mathscr{SM}, \mathscr{OD})$ and a structure $\mathscr{D}$ denotes a set $[\![\mathcal{M}]\!]$ of (initial and consecutive) **computations** of the form

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

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

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

Definition. Let $\mathcal{M} = (\mathscr{CD}, \mathscr{SM}, \mathscr{OD})$ be a UML model and $\mathscr{D}$ a structure. Then

$$\mathcal{L}(\mathcal{M}) := \{(\sigma_i, cons_i, Snd_i)_{i \in \mathbb{N}_0} \in (\Sigma_{\mathscr{S}}^{\mathscr{D}} \times \tilde{A})^\omega \mid \exists\, (\varepsilon_i, u_i)_{i \in \mathbb{N}_0} : (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(cons_0, Snd_0)} (\sigma_1, \varepsilon_1) \dots \in [\![\mathcal{M}]\!]\}$$

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

– 19 – 2014-01-29 – Smodellang –

Example: The Language of a Model

$$\mathcal{L}(\mathcal{M}) := \{(\sigma_i, cons_i, Snd_i)_{i \in \mathbb{N}_0} \in (\Sigma_{\mathscr{S}}^{\mathscr{D}} \times \tilde{A})^\omega \mid \exists\, (\varepsilon_i, u_i)_{i \in \mathbb{N}_0} : (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(cons_0, Snd_0)} (\sigma_1, \varepsilon_1) \dots \in [\![\mathcal{M}]\!]\}$$

– 19 – 2014-01-29 – Smodellang –

Signal and Attribute Expressions

- Let $\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr, \mathscr{E})$ be a signature and X a set of logical variables,

- The signal and attribute expressions $Expr_{\mathscr{S}}(\mathscr{E}, X)$ are defined by the grammar:

$$\psi ::= \textit{true} \mid expr \mid E^!_{x,y} \mid E^?_{x,y} \mid \neg\psi \mid \psi_1 \vee \psi_2,$$

where $expr : Bool \in Expr_{\mathscr{S}}$, $E \in \mathscr{E}$, $x, y \in X$.

Satisfaction of Signal and Attribute Expressions

- Let $(\sigma, cons, Snd) \in \Sigma^{\mathscr{D}}_{\mathscr{S}} \times \tilde{A}$ be a triple consisting of **system state**, **consume set**, and **send set**.
- Let $\beta : X \to \mathscr{D}(\mathscr{C})$ be a valuation of the logical variables.

Then

- $(\sigma, cons, Snd) \models_\beta$ *true*
- $(\sigma, cons, Snd) \models_\beta \neg\psi$ if and only if not $(\sigma, cons, Snd) \models_\beta \psi$
- $(\sigma, cons, Snd) \models_\beta \psi_1 \vee \psi_2$ if and only if $(\sigma, cons, Snd) \models_\beta \psi_1$ or $(\sigma, cons, Snd) \models_\beta \psi_2$
- $(\sigma, cons, Snd) \models_\beta expr$ if and only if $I[\![expr]\!](\sigma, \beta) = 1$
- $(\sigma, cons, Snd) \models_\beta E^!_{x,y}$ if and only if $\exists\, \vec{d} \bullet (\beta(x), (E, \vec{d}), \beta(y)) \in Snd$
- $(\sigma, cons, Snd) \models_\beta E^?_{x,y}$ if and only if $\exists\, \vec{d} \bullet (\beta(x), (E, \vec{d}), \beta(y)) \in cons$

Observation: semantics of models **keeps track** of sender and receiver at sending and consumption time. We disregard the event identity.
Alternative: keep track of event identities.

TBA over Signature

Definition. A TBA

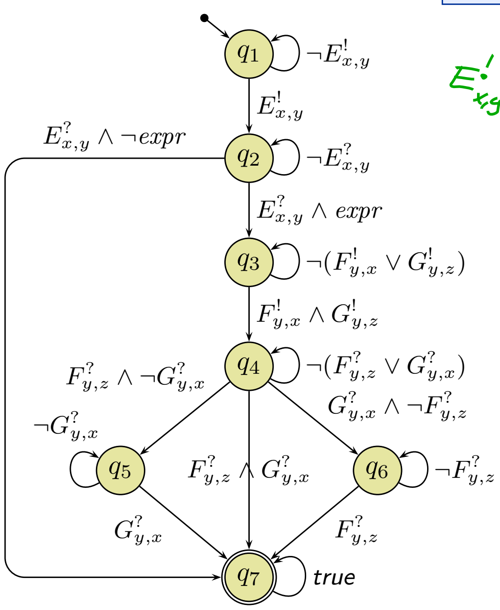
$$\mathcal{B} = (Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \rightarrow, Q_F)$$

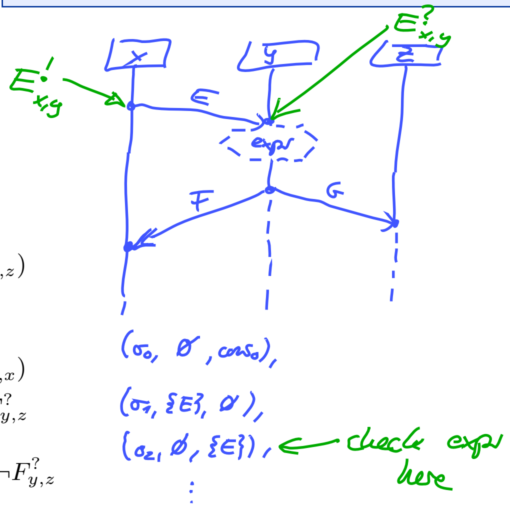
where $Expr_{\mathcal{B}}(X)$ is the set of **signal and attribute expressions** $Expr_{\mathscr{S}}(\mathscr{E}, X)$ over signature $\mathscr{S}$ is called **TBA over $\mathscr{S}$**.

- Any word over $\mathscr{S}$ and $\mathscr{D}$ is then a word for $\mathcal{B}$.
  (By the satisfaction relation defined on the previous slide; $\mathscr{D}(X) = \mathscr{D}(\mathscr{C})$.)
- Thus a TBA over $\mathscr{S}$ accepts words of models with signature $\mathscr{S}$.
  (By the previous definition of TBA.)

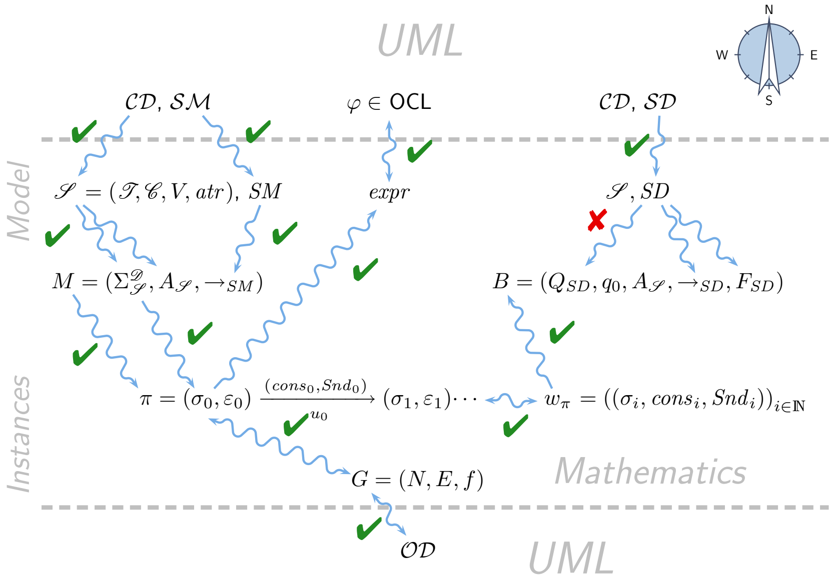
TBA over Signature Examp

$(\sigma, cons, Snd) \models_\beta expr$ iff $I[\![expr]\!](\sigma, \beta) = 1$;
$(\sigma, cons, Snd) \models_\beta E^!_{x,y}$ iff $(\beta(x), (E, \vec{d}), \beta(y)) \in Snd$





Course Map



Live Sequence Charts Semantics

TBA-based Semantics of LSCs

Plan:

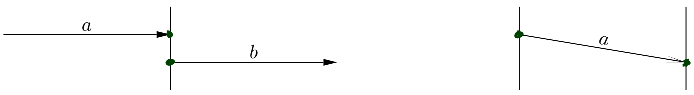
- Given an LSC L with body

$$(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}),$$

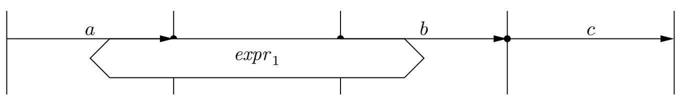
- construct a TBA $\mathcal{B}_L$, and
- define $\mathcal{L}(L)$ **in terms of** $\mathcal{L}(\mathcal{B}_L)$,
  in particular taking activation condition and activation mode into account.
- Then $\mathcal{M} \models L$ (universal) if and only if $\mathcal{L}(\mathcal{M}) \subseteq \mathcal{L}(L)$.

Recall: Intuitive Semantics

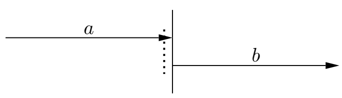
(i) **Strictly After:**



(ii) **Simultaneously:** (simultaneous region)

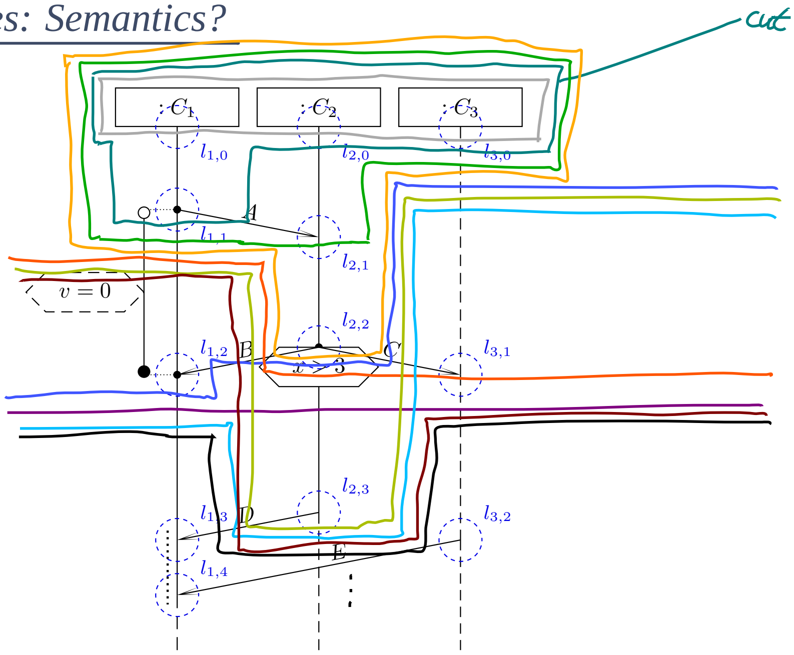


(iii) **Explicitly Unordered:** (co-region)



Intuition: A computation path **violates** an LSC if the occurrence of some events doesn't adhere to the partial order obtained as the **transitive closure** of (i) to (iii).

Examples: Semantics?



Formal LSC Semantics: It's in the Cuts!

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

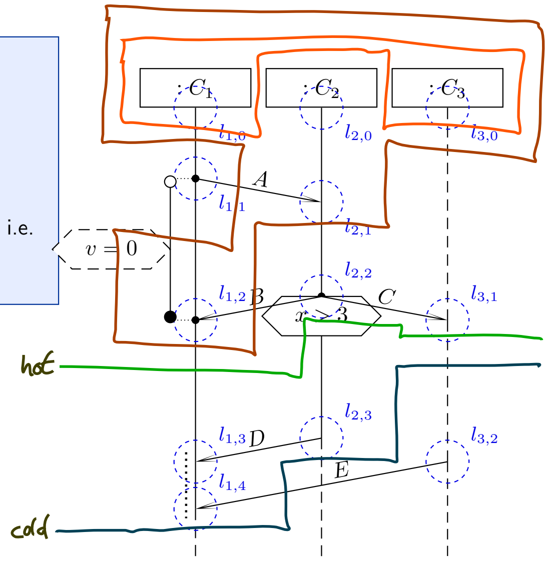
- it is **downward closed**, i.e.
$$\forall l, l' : l' \in C \wedge l \preceq l' \implies l \in C,$$
- it is **closed** under **simultaneity**, i.e.
$$\forall l, l' : l' \in C \wedge l \sim l' \implies l \in C, \text{ and}$$
- it comprises at least **one location per instance line**, i.e.
$$\forall i \in I\ \exists l \in C : i_l = i.$$

A cut C is called **hot**, denoted by $\theta(C) = \mathsf{hot}$, if and only if at least one of its maximal elements is hot, i.e. if
$$\exists l \in C : \theta(l) = \mathsf{hot} \wedge \nexists l' \in C : l \prec l'$$
Otherwise, C is called **cold**, denoted by $\theta(C) = \mathsf{cold}$.

Examples: Cut or Not Cut? Hot/Cold?

(i) **non-empty** set $\emptyset \neq C \subseteq \mathscr{L}$,
(ii) **downward closed**, i.e.
$\forall\, l, l' : l' \in C \wedge l \preceq l' \implies l \in C$
(iii) **closed** under **simultaneity**, i.e.
$\forall\, l, l' : l' \in C \wedge l \sim l' \implies l \in C$
(iv) at least **one location per instance line**, i.e.
$\forall\, i \in I\; \exists\, l \in C : i_l = i,$

- $C_0 = \emptyset$
- $C_1 = \{l_{1,0}, l_{2,0}, l_{3,0}\}$
- $C_2 = \{l_{1,1}, l_{2,1}, l_{3,0}\}$
- $C_3 = \{l_{1,0}, l_{1,1}\}$
- $C_4 = \{l_{1,0}, l_{1,1}, l_{2,0}, l_{3,0}\}$
- $C_5 = \{l_{1,0}, l_{1,1}, l_{2,0}, l_{2,1}, l_{3,0}\}$
- $C_6 = \mathscr{L} \setminus \{l_{1,3}, l_{2,3}\}$
- $C_7 = \mathscr{L}$



A Successor Relation on Cuts

The partial order of $(\mathscr{L}, \preceq)$ and the simultaneity relation "$\sim$" induce a **direct successor relation** on cuts of $\mathscr{L}$ as follows:

Definition. Let $C, C' \subseteq \mathscr{L}$ bet cuts of an LSC body with locations $(\mathscr{L}, \preceq)$ and messages Msg.
C' is called **direct successor** of C **via fired-set** F, denoted by $C \rightsquigarrow_F C'$, if and only if

- $F \neq \emptyset$,
- $C' \setminus C = F$,
- for each message reception in F, the corresponding sending is already in C,
$$\forall\, (l, E, l') \in \mathsf{Msg} : l' \in F \implies l \in C, \text{ and}$$
- locations in F, that lie on the same instance line, are pairwise unordered, i.e.
$$\forall\, l, l' \in F : l \neq l' \wedge i_l = i_{l'} \implies l \not\preceq l' \wedge l' \not\preceq l$$

Properties of the Fired-set

$C \rightsquigarrow_F C'$ if and only if
- $F \neq \emptyset$,
- $C' \setminus C = F$,
- $\forall (l, E, l') \in \mathsf{Msg} : l' \in F \implies l \in C$, and
- $\forall l, l' \in F : l \neq l' \wedge i_l = i_{l'} \implies l \not\preceq l' \wedge l' \not\preceq l$

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

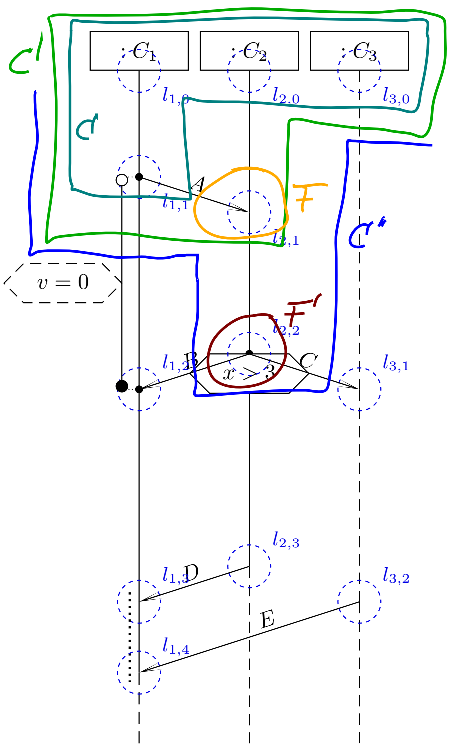
$$\forall l' \in F\ \exists l \in C : l \prec l' \wedge \nexists l'' \in C : l' \prec l'' \prec l$$

Successor Cut Examples

(i) $F \neq \emptyset$, (ii) $C' \setminus C = F$,
(iii) $\forall (l, E, l') \in \mathsf{Msg} : l' \in F \implies l \in C$, and
(iv) $\forall l, l' \in F : l \neq l' \wedge i_l = i_{l'} \implies l \not\preceq l' \wedge l' \not\preceq l$

$C' \rightsquigarrow_F C'$

$C' \rightsquigarrow_{F'} C''$



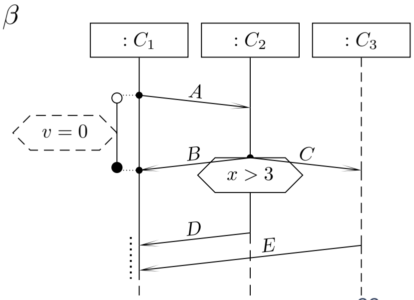
Idea: Accept Timed Words by Advancing the Cut

- Let $w = (\sigma_0, cons_0, Snd_0), (\sigma_1, cons_1, Snd_1), (\sigma_2, cons_2, Snd_2), \dots$ be a word of a UML model and β a valuation of $I \cup \{self\}$.
- **Intuitively** (and for now **disregarding** cold conditions), an LSC body $(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv})$ is **supposed** to **accept** w if and only if there exists a sequence

$$C_0 \rightsquigarrow_{F_1} C_1 \rightsquigarrow_{F_2} C_2 \cdots \rightsquigarrow_{F_n} C_n$$

and indices $0 = i_0 < i_1 < \cdots < i_n$ such that for all $0 \leq j < n$,

  - for all $i_j \leq k < i_{j+1}$, $(\sigma_k, cons_k, Snd_k)$, β satisfies the **hold condition** of C_j,
  - $(\sigma_{i_j}, cons_{i_j}, Snd_{i_j})$, β satisfies the **transition condition** of F_j,
  - C_n is cold,
  - for all $i_n < k$, $(\sigma_k, cons_{i_j}, Snd_{i_j})$, β satisfies the **hold condition** of C_n.



Language of LSC Body

The **language** of the body

$$(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv})$$

of LSC L is the language of the TBA

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

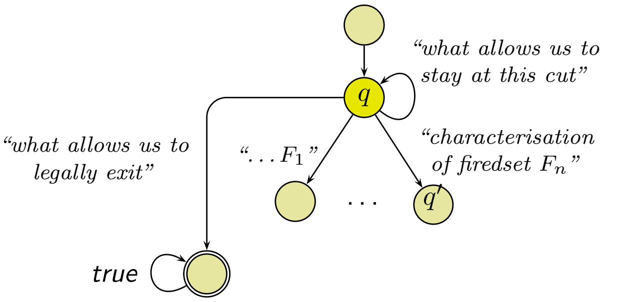
with

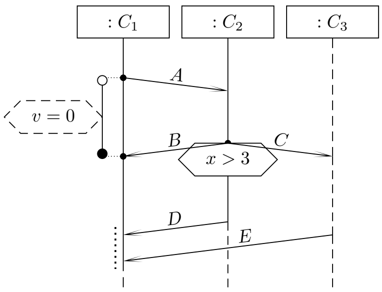
- $Expr_{\mathcal{B}}(X) = Expr_{\mathscr{S}}(\mathscr{S}, X)$
- Q is the set of cuts of $(\mathscr{L}, \preceq)$, q_{ini} is the **instance heads** cut,
- $F = \{C \in Q \mid \theta(C) = \mathsf{cold}\}$ is the set of cold cuts of $(\mathscr{L}, \preceq)$,
- $\rightarrow$ as defined in the following, consisting of
  - **loops** (q, ψ, q),
  - **progress transitions** (q, ψ, q') corresponding to $q \rightsquigarrow_F q'$, and
  - **legal exits** $(q, \psi, \mathscr{L})$.

Language of LSC Body: Intuition

$\mathcal{B}_L = (Expr_\mathcal{B}(X), X, Q, q_{ini}, \rightarrow, Q_F)$ with

- $Expr_\mathcal{B}(X) = Expr_\mathscr{S}(\mathscr{S}, X)$
- Q is the set of cuts of $(\mathscr{L}, \preceq)$, q_{ini} is the **instance heads** cut,
- $F = \{C \in Q \mid \theta(C) = \text{cold}\}$ is the set of cold cuts,
- $\rightarrow$ consists of
  - **loops** (q, ψ, q),
  - **progress transitions** (q, ψ, q') corresponding to $q \rightsquigarrow_F q'$, and
  - **legal exits** $(q, \psi, \mathscr{L})$.





Step I: Only Messages

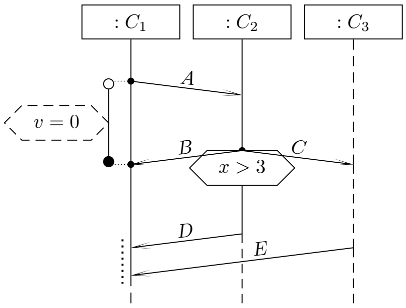
Some Helper Functions

- **Message-expressions of a location**:

$$\mathscr{E}(l) := \{E^!_{i_l,i_{l'}} \mid (l,E,l') \in \mathsf{Msg}\} \cup \{E^?_{i_{l'},i_l} \mid (l',E,l) \in \mathsf{Msg}\},$$

$$\mathscr{E}(\{l_1,\dots,l_n\}) := \mathscr{E}(l_1) \cup \dots \cup \mathscr{E}(l_n).$$

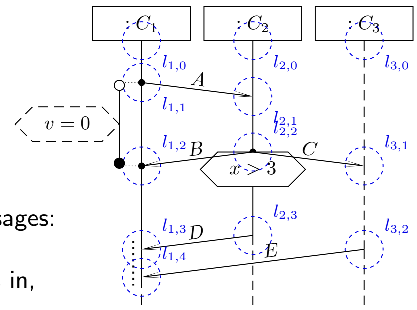
$$\bigvee \emptyset := \textit{true}; \bigvee\{E1^!_{i_{11},i_{12}}, \dots Fk^?_{i_{k1},i_{k2}}, \dots\} := \bigvee_{1\le j<k} Ej^!_{i_{j1},i_{j2}} \vee \bigvee_{k\le j} Fj^?_{i_{j1},i_{j2}}$$



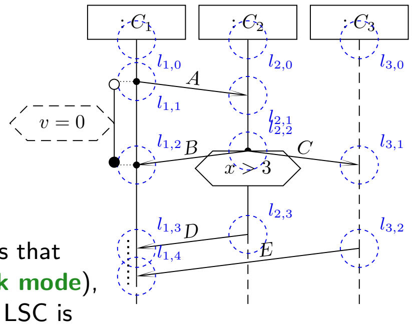
Loops

- How long may we **legally** stay at a cut q?
- **Intuition**: those $(\sigma_i, cons_i, Snd_i)$ are allowed to fire the self-loop (q, ψ, q) where
  - $cons_i \cup Snd_i$ comprises only irrelevant messages:
    - **weak mode**: no message from a direct successor cut is in,
    - **strict mode**: no message occurring in the LSC is in,
  - sigma_i satisfies the local invariants active at q

  And nothing else.
- **Formally**: Let $F := F_1 \cup \dots \cup F_n$ be the union of the firedsets of q.
  - $\psi := \underbrace{\neg(\bigvee \mathscr{E}(F))}_{=\textit{true} \text{ if } F=\emptyset} \wedge \bigwedge \psi(q).$



Progress



- When do we move from q to q'?
- **Intuition**: those $(\sigma_i, cons_i, Snd_i)$ fire the progress transition (q, ψ, q') for which there exists a firedset F such that $q \rightsquigarrow_F q'$ and
  - $cons_i \cup Snd_i$ comprises exactly the messages that distinguish F from other firedsets of q (**weak mode**), and in addition no message occurring in the LSC is in $cons_i \cup Snd_i$ (**strict mode**),
  - sigma_i satisfies the local invariants and conditions relevant at q
- **Formally**: Let $F, F_1, \dots, F_n$ be the firedsets of q and let $q \rightsquigarrow_F q'$ (unique).
  - $\psi := \bigwedge \mathscr{E}(F) \wedge \neg(\bigvee(\mathscr{E}(F_1) \cup \dots \cup \mathscr{E}(F_n)) \setminus \mathscr{E}(F)) \wedge \bigwedge \psi(q, q')$.

Step II: Conditions and Local Invariants

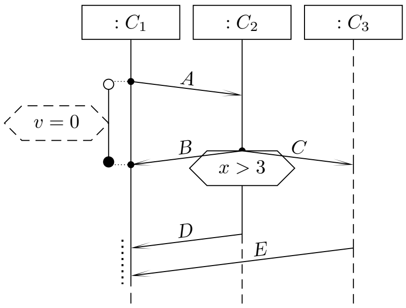
Some More Helper Functions

- **Constraints** relevant **at** cut q:

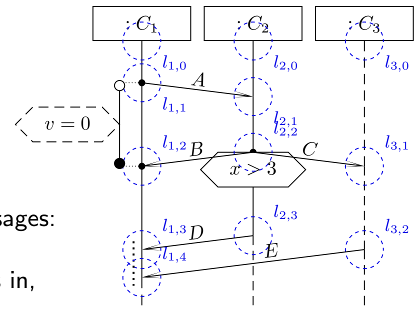
$$\psi_\theta(q) = \{\psi \mid \exists\, l \in q, l' \notin q \mid (l, \psi, \theta, l') \in \mathsf{LocInv} \vee (l', \psi, \theta, l) \in \mathsf{LocInv}\},$$

$$\psi(q) = \psi_{\mathsf{hot}}(q) \cup \psi_{\mathsf{cold}}(q)$$

$$\bigwedge \emptyset := \mathit{false}; \quad \bigwedge\{\psi_1, \dots, \psi_n\} := \bigwedge_{1 \le i \le n} \psi_i$$



Loops with Conditions



- How long may we **legally** stay at a cut q?
- **Intuition**: those $(\sigma_i, cons_i, Snd_i)$ are allowed to fire the self-loop (q, ψ, q) where
  - $cons_i \cup Snd_i$ comprises only irrelevant messages:
    - **weak mode**:
      no message from a direct successor cut is in,
    - **strict mode**:
      no message occurring in the LSC is in,
  - σ_i satisfies the local invariants active at q

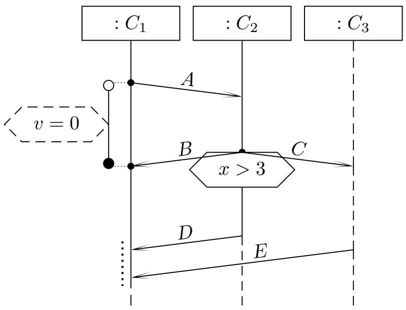
  And nothing else.
- **Formally**: Let $F := F_1 \cup \dots \cup F_n$ be the union of the firedsets of q.
  - $\psi := \underbrace{\neg(\bigvee \mathscr{E}(F))}_{= \mathit{true} \text{ if } F = \emptyset} \wedge \bigwedge \psi(q).$

Even More Helper Functions

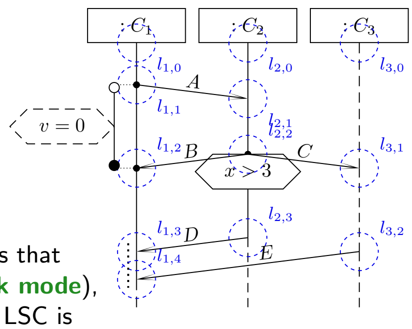
- **Constraints** relevant when moving from q **to** cut q':

$$\psi_\theta(q,q') = \{\psi \mid \exists\, L \subseteq \mathscr{L} \mid (L,\psi,\theta) \in \mathsf{Cond} \wedge L \cap (q' \setminus q) \neq \emptyset\}$$
$$\cup\, \psi_\theta(q')$$
$$\setminus \{\psi \mid \exists\, l \in q' \setminus q, l' \in \mathscr{L} \mid (l, \circ, expr, \theta, l') \in \mathsf{LocInv} \vee (l', expr, \theta, \circ, l) \in \mathsf{LocInv}\}$$
$$\cup \{\psi \mid \exists\, l \in q' \setminus q, l' \in \mathscr{L} \mid (l, \bullet, expr, \theta, l') \in \mathsf{LocInv} \vee (l', expr, \theta, \bullet, l) \in \mathsf{LocInv}\}$$

$$\psi(q,q') = \psi_{\mathsf{hot}}(q,q') \cup \psi_{\mathsf{cold}}(q,q')$$



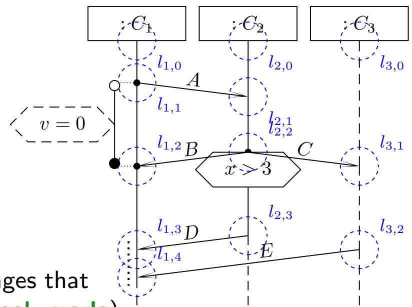
Progress with Conditions



- When do we move from q to q'?
- **Intuition**: those $(\sigma_i, cons_i, Snd_i)$ fire the progress transition (q, ψ, q') for which there exists a firedset F such that $q \rightsquigarrow_F q'$ and
  - $cons_i \cup Snd_i$ comprises exactly the messages that distinguish F from other firedsets of q (**weak mode**), and in addition no message occurring in the LSC is in $cons_i \cup Snd_i$ (**strict mode**),
  - σ_i satisfies the local invariants and conditions relevant at q'.
- **Formally**: Let $F, F_1, \dots, F_n$ be the firedsets of q and let $q \rightsquigarrow_F q'$ (unique).
  - $\psi := \bigwedge \mathscr{E}(F) \wedge \neg\big(\bigvee\big(\mathscr{E}(F_1) \cup \dots \cup \mathscr{E}(F_n)\big) \setminus \mathscr{E}(F)\big) \wedge \bigwedge \psi(q,q')$.

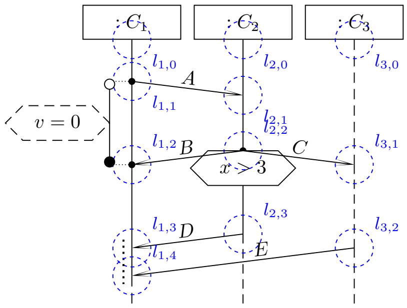
Step III: Cold Conditions and Cold Local Invariants

Legal Exits



- When do we take a legal exit from q?
- **Intuition**: those $(\sigma_i, cons_i, Snd_i)$ fire the legal exit transition $(q, \psi, \mathscr{L})$
  - for which there exists a firedset F and some q' such that $q \rightsquigarrow_F q'$ and
    - $cons_i \cup Snd_i$ comprises exactly the messages that distinguish F from other firedsets of q (**weak mode**), and in addition no message occurring in the LSC is in $cons_i \cup Snd_i$ (**strict mode**) and
    - at least one cold condition or local invariant relevant when moving to q' is violated, or
  - for which there is no matching firedset and at least one cold local invariant relevant at q is violated.
- **Formally**: Let $F_1, \ldots, F_n$ be the firedsets of q with $q \rightsquigarrow_{F_i} q_i'$.
  - $\psi := \bigvee_{i=1}^{n} \bigwedge \mathscr{E}(F_i) \wedge \neg\big(\bigvee(\mathscr{E}(F_1) \cup \cdots \cup \mathscr{E}(F_n)) \setminus \mathscr{E}(F_i)\big) \wedge \bigvee \psi_{\mathsf{cold}}(q, q_i')$
    $\vee \neg(\bigvee \mathscr{E}(F_i)) \wedge \bigvee \psi_{\mathsf{cold}}(q)$

Example



Finally: The LSC Semantics

A **full LSC** L consist of

- a **body** $(I, (\mathscr{L}, \preceq), \sim, \mathscr{S}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv})$,
- an **activation condition** (here: event) $ac = E^{?}_{i_1,i_2}$, $E \in \mathscr{E}$, $i_1, i_2 \in I$,
- an **activation mode**, either **initial** or **invariant**,
- a **chart mode**, either **existential** (cold) or **universal** (hot).

A set W of words over $\mathscr{S}$ and $\mathscr{D}$ **satisfies** L, denoted $W \models L$, iff L

- **universal** (= hot), **initial**, and

  $$\forall\, w \in W\; \forall\, \beta : I \to \mathrm{dom}(\sigma(w^0)) \bullet w \text{ activates } L \implies w \in \mathcal{L}_\beta(\mathcal{B}_L).$$
- **existential** (= cold), **initial**, and

  $$\exists\, w \in W\; \exists\, \beta : I \to \mathrm{dom}(\sigma(w^0)) \bullet w \text{ activates } L \wedge w \in \mathcal{L}_\beta(\mathcal{B}_L).$$
- **universal** (= hot), **invariant**, and

  $$\forall\, w \in W\; \forall\, k \in \mathbb{N}_0\; \forall\, \beta : I \to \mathrm{dom}(\sigma(w^k)) \bullet w/k \text{ activates } L \implies w/k \in \mathcal{L}_\beta(\mathcal{B}_L).$$
- **existential** (= cold), **invariant**, and

  $$\exists\, w \in W\; \exists\, k \in \mathbb{N}_0\; \exists\, \beta : I \to \mathrm{dom}(\sigma(w^k)) \bullet w/k \text{ activates } L \wedge w/k \in \mathcal{L}_\beta(\mathcal{B}_L).$$

Back to UML: Interactions

Model Consistency wrt. Interaction

- We assume that the set of interactions $\mathscr{I}$ is partitioned into two (possibly empty) sets of **universal** and **existential** interactions, i.e.

$$\mathscr{I} = \mathscr{I}_\forall \,\dot{\cup}\, \mathscr{I}_\exists.$$

Definition. A model

$$\mathcal{M} = (\mathscr{C\!D}, \mathscr{S\!M}, \mathscr{O\!D}, \mathscr{I})$$

is called **consistent** (more precise: the constructive description of behaviour is consistent with the reflective one) if and only if

$$\forall \mathcal{I} \in \mathscr{I}_\forall : \mathcal{L}(\mathcal{M}) \subseteq \mathcal{L}(\mathcal{I})$$

and

$$\forall \mathcal{I} \in \mathscr{I}_\exists : \mathcal{L}(\mathcal{M}) \cap \mathcal{L}(\mathcal{I}) \neq \emptyset.$$

Interactions as Reflective Description

- In UML, reflective (temporal) descriptions are subsumed by **interactions**.
- A UML model $\mathcal{M} = (\mathscr{CD}, \mathscr{SM}, \mathscr{OD}, \mathscr{I})$ has a set of interactions $\mathscr{I}$.
- An interaction $\mathcal{I} \in \mathscr{I}$ can be (OMG claim: equivalently) **diagrammed** as
  - **sequence diagram**, **timing diagram**, or
  - **communication diagram** (formerly known as collaboration diagram).

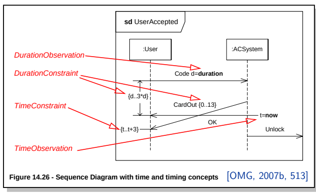


Figure 14.26 - Sequence Diagram with time and timing concepts [OMG, 2007b, 513]

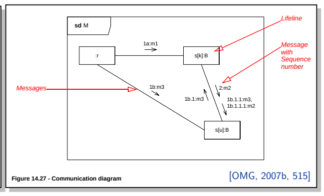


Figure 14.27 - Communication diagram [OMG, 2007b, 515]

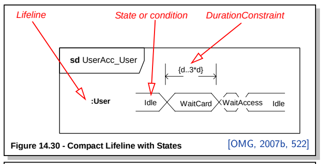


Figure 14.30 - Compact Lifeline with States [OMG, 2007b, 522]

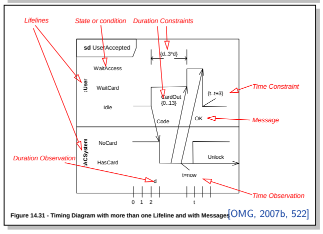


Figure 14.31 - Timing Diagram with more than one Lifeline and with Messages [OMG, 2007b, 522]

– 19 – 2014-01-29 – Sinteract –

Interactions as Reflective Description

- In UML, reflective (temporal) descriptions are subsumed by **interactions**.
- A UML model $\mathcal{M} = (\mathscr{CD}, \mathscr{SM}, \mathscr{OD}, \mathscr{I})$ has a set of interactions $\mathscr{I}$.
- An interaction $\mathcal{I} \in \mathscr{I}$ can be (OMG claim: equivalently) **diagrammed** as
  - **sequence diagram**, **timing diagram**, or
  - **communication diagram** (formerly known as collaboration diagram).

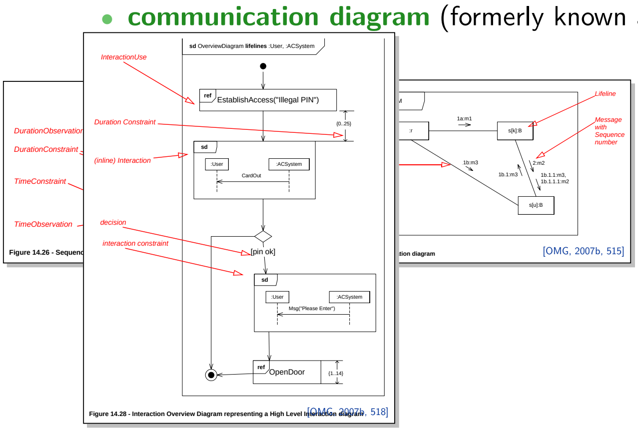


Figure 14.28 - Interaction Overview Diagram representing a High Level Interaction diagram [OMG, 2007b, 518]

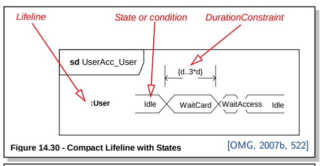


Figure 14.30 - Compact Lifeline with States [OMG, 2007b, 522]

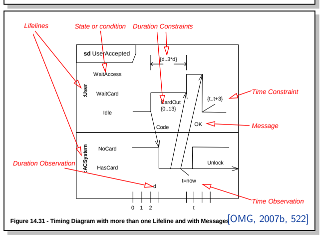
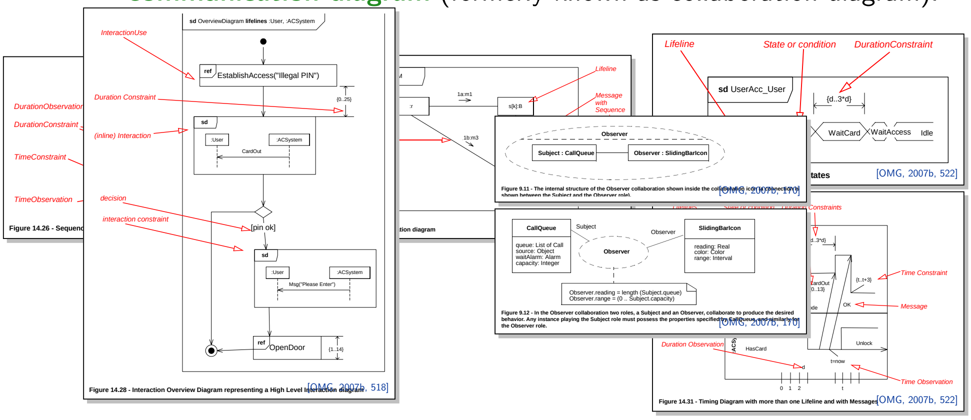


Figure 14.31 - Timing Diagram with more than one Lifeline and with Messages [OMG, 2007b, 522]

– 19 – 2014-01-29 – Sinteract –

Interactions as Reflective Description

- In UML, reflective (temporal) descriptions are subsumed by **interactions**.
- A UML model $\mathcal{M} = (\mathscr{CD}, \mathscr{SM}, \mathscr{OD}, \mathscr{I})$ has a set of interactions $\mathscr{I}$.
- An interaction $\mathcal{I} \in \mathscr{I}$ can be (OMG claim: equivalently) **diagrammed** as
  - **sequence diagram**, **timing diagram**, or
  - **communication diagram** (formerly known as collaboration diagram).



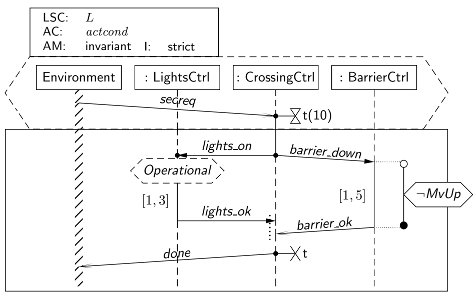
Why Sequence Diagrams?

Most Prominent: Sequence Diagrams — with **long history**:

- **Message Sequence Charts**, standardized by the ITU in different versions, often accused to lack a formal semantics.
- **Sequence Diagrams** of UML 1.x

Most severe **drawbacks** of these formalisms:

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



Thus: Live Sequence Charts

- **SDs of UML 2.x** address **some** issues, yet the standard exhibits unclarities and even contradictions [Harel and Maoz, 2007, Störrle, 2003]
- For the lecture, we consider **Live Sequence Charts** (LSCs) [Damm and Harel, 2001, Klose, 2003, Harel and Marelly, 2003], who have a common fragment with UML 2.x SDs [Harel and Maoz, 2007]
- **Modelling guideline**: stick to that fragment.

Side Note: Protocol Statemachines

Same direction: **call orders** on operations

- **"for each C instance, method $f()$ shall only be called after $g()$ but before $h()$"**

Can be formalised with protocol state machines.

References

References

[Damm and Harel, 2001] Damm, W. and Harel, D. (2001). LSCs: Breathing life into Message Sequence Charts. *Formal Methods in System Design*, 19(1):45–80.

[Harel and Gery, 1997] Harel, D. and Gery, E. (1997). Executable object modeling with statecharts. *IEEE Computer*, 30(7):31–42.

[Harel and Maoz, 2007] Harel, D. and Maoz, S. (2007). Assert and negate revisited: Modal semantics for UML sequence diagrams. *Software and System Modeling (SoSyM)*. To appear. (Early version in SCESM'06, 2006, pp. 13-20).

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

[Klose, 2003] Klose, J. (2003). *LSCs: A Graphical Formalism for the Specification of Communication Behavior*. PhD thesis, Carl von Ossietzky Universität Oldenburg.

[OMG, 2007a] OMG (2007a). Unified modeling language: Infrastructure, version 2.1.2. Technical Report formal/07-11-04.

[OMG, 2007b] OMG (2007b). Unified modeling language: Superstructure, version 2.1.2. Technical Report formal/07-11-02.

[Störrle, 2003] Störrle, H. (2003). Assert, negate and refinement in UML-2 interactions. In Jürjens, J., Rumpe, B., France, R., and Fernandez, E. B., editors, *CSDUML 2003*, number TUM-I0323. Technische Universität München.