Software Design, Modelling and Analysis in UML

Lecture 19: Live Sequence Charts III

2017-01-26

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

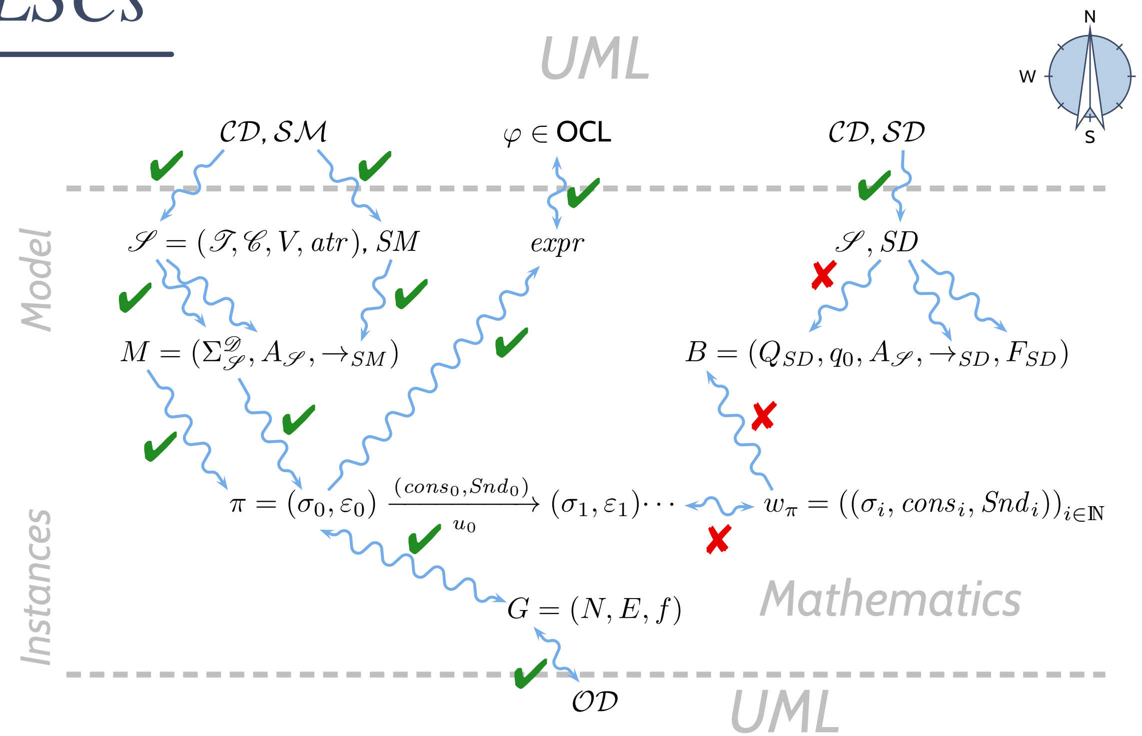
Albert-Ludwigs-Universität Freiburg, Germany

Content

- Live Sequence Charts
  - Semantics
    - TBA Construction for LSC Body
      - Cuts, Firedsets
      - Signal / Attribute Expressions
      - Loop / Progress Conditions
    - Excursion: Büchi Automata
    - Language of a Model
    - Full LSCs
      - Existential and Universal
      - Pre-Charts
      - Forbidden Scenarios
    - LSCs and Tests

Live Sequence Charts — Semantics

TBA-based Semantics of LSCs



Plan:

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

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

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

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

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

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

Live Sequence Charts — TBA Construction

Formal LSC Semantics: It's in the Cuts!

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

- it is **downward closed**, i.e. $\forall\, l, l' \bullet l' \in C \wedge l \preceq l' \implies l \in C$,
- it is **closed** under **simultaneity**, i.e.

$$\forall\, l, l' \bullet l' \in C \wedge l \sim l' \implies l \in C, \text{ and}$$

- it comprises at least **one location per instance line**, i.e.

$$\forall\, i \in I\, \exists\, l \in C \bullet i_l = i.$$

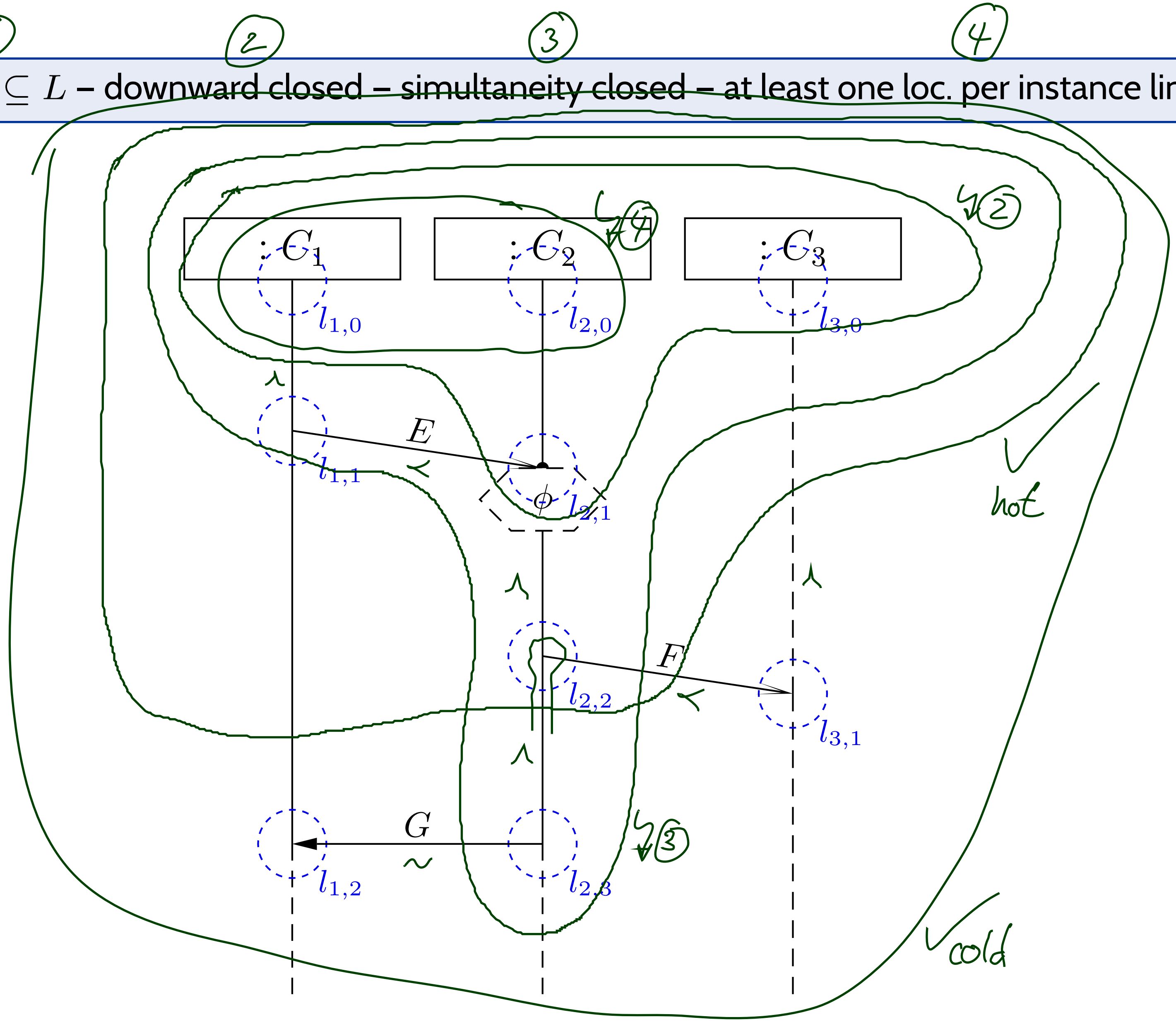
The **temperature function** is extended to cuts as follows:

$$\Theta(C) = \begin{cases} \mathsf{hot} & \text{, if } \exists\, l \in C \bullet (\nexists\, l' \in C \bullet l \prec l') \wedge \Theta(l) = \mathsf{hot} \\ \mathsf{cold} & \text{, otherwise} \end{cases}$$

that is, C is **hot** if and only if at least one of its maximal elements is hot.

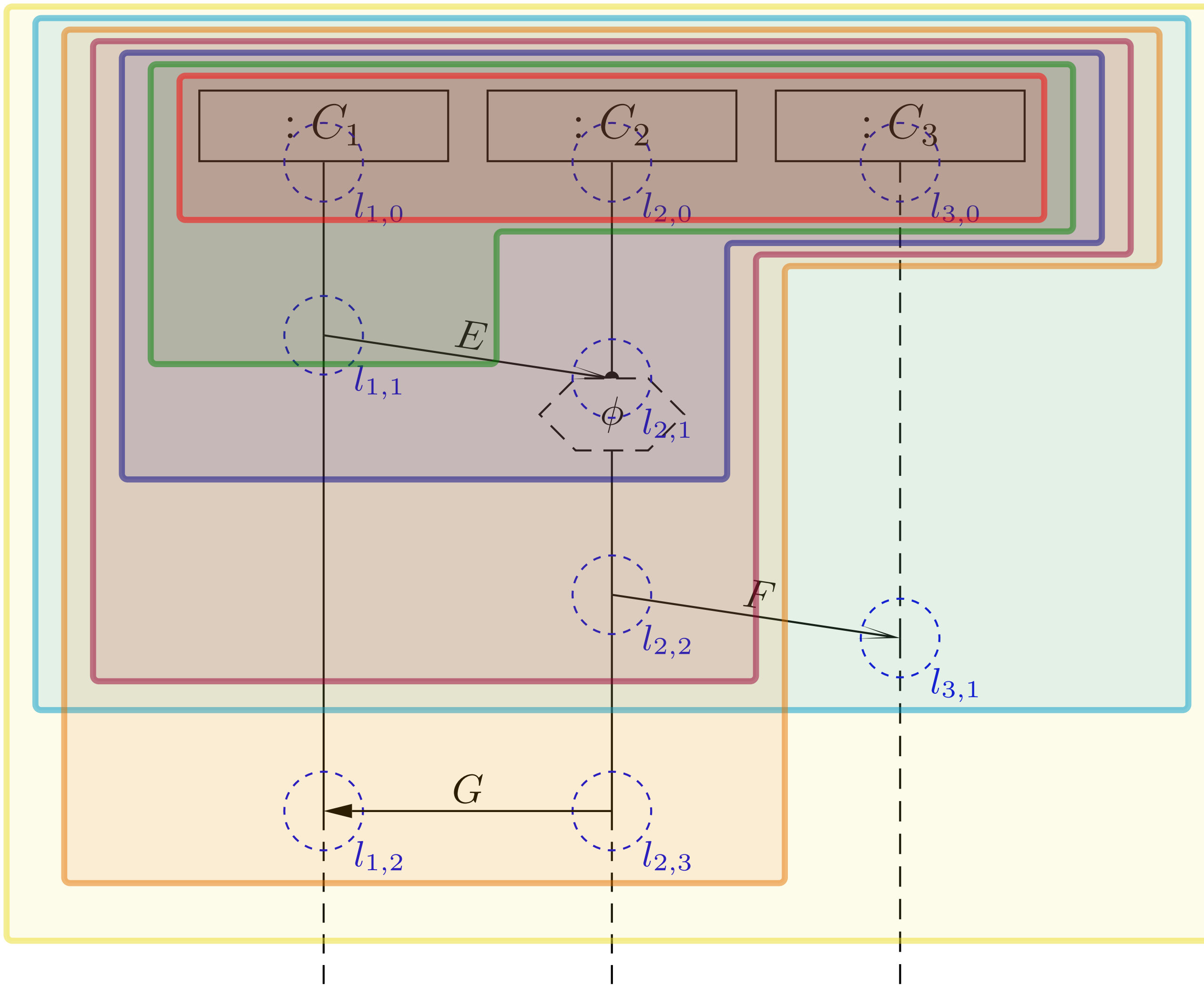
Cut Examples

$\emptyset \neq C \subseteq L$ – downward closed – simultaneity closed – at least one loc. per instance line



Cut Examples

$\emptyset \neq C \subseteq L$ – downward closed – simultaneity closed – at least one loc. per instance line



A Successor Relation on Cuts

The partial order "$\preceq$" and the simultaneity relation "$\sim$" of locations induce a **direct successor relation** on cuts of an LSC body as follows:

Definition.
Let $C \subseteq L$ bet a cut of LSC body $((L, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta)$.

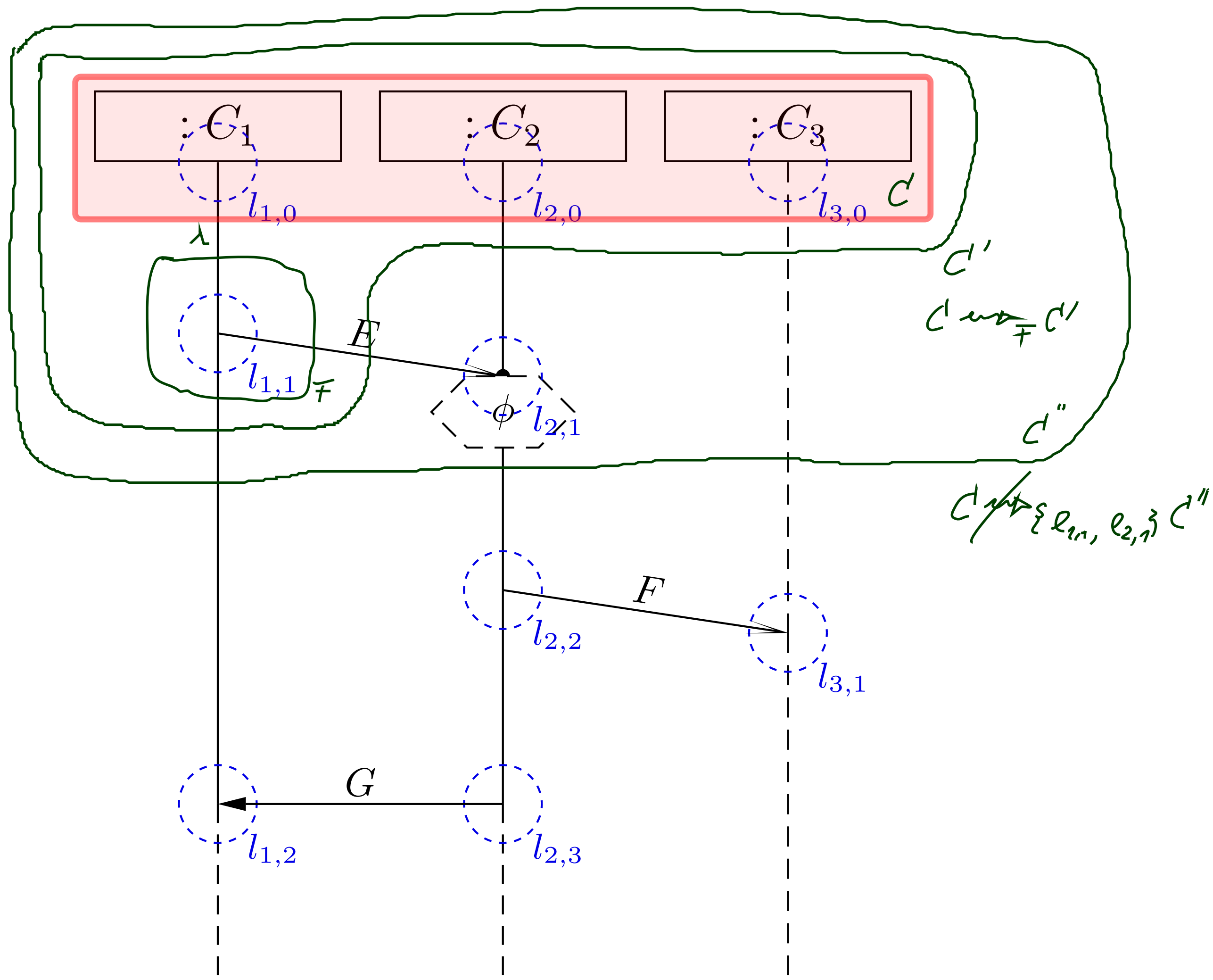
A set $\emptyset \neq F \subseteq L$ of locations is called fired-set F of cut C if and only if

- $C \cap F = \emptyset$ and $C \cup F$ is a cut, i.e. F is closed under simultaneity,
- all locations in F are direct $\prec$-successors of the front of C, i.e.
  $$\forall\, l \in F\, \exists\, l' \in C \bullet l' \prec l \wedge (\nexists\, l'' \in C \bullet l' \prec l'' \prec l),$$
- locations in F, that lie on the same instance line, are pairwise unordered, i.e.
  $$\forall\, l \neq l' \in F \bullet (\exists\, I \in \mathcal{I} \bullet \{l, l'\} \subseteq I) \implies l \not\preceq l' \wedge l' \not\preceq l,$$
- for each asynchronous (!) message reception in F,
  the corresponding sending is already in C,
  $$\forall\, (l, E, l') \in \mathsf{Msg} \bullet l' \in F \implies l \in C.$$

The cut $C' = C \cup F$ is called direct successor of C via F, denoted by $C \rightsquigarrow_F C'$.

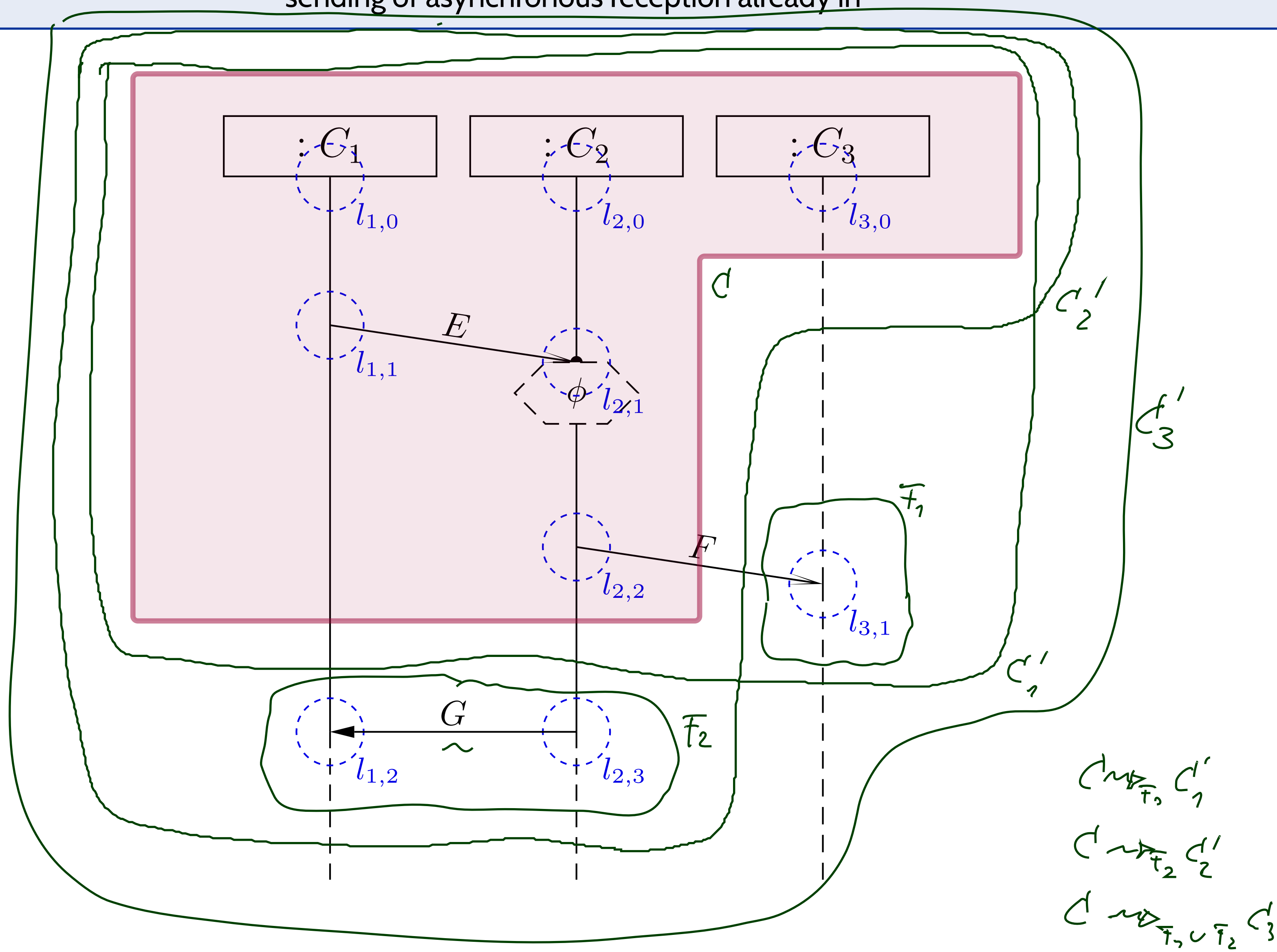
Successor Cut Example

$C \cap F = \emptyset$ – $C \cup F$ is a cut – only direct $\prec$-successors – same instance line on front pairwise unordered – sending of asynchronous reception already in

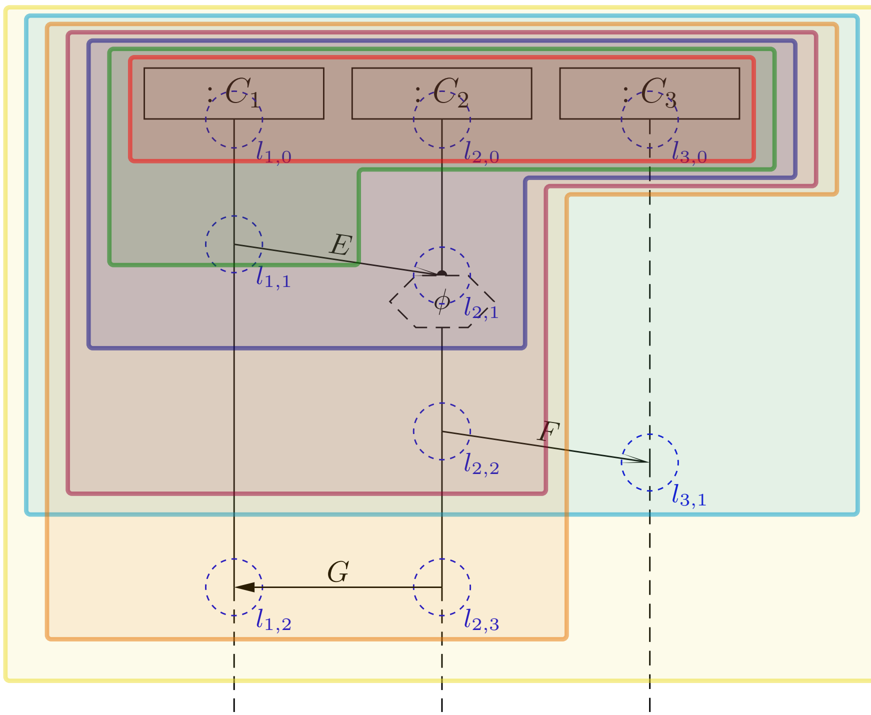


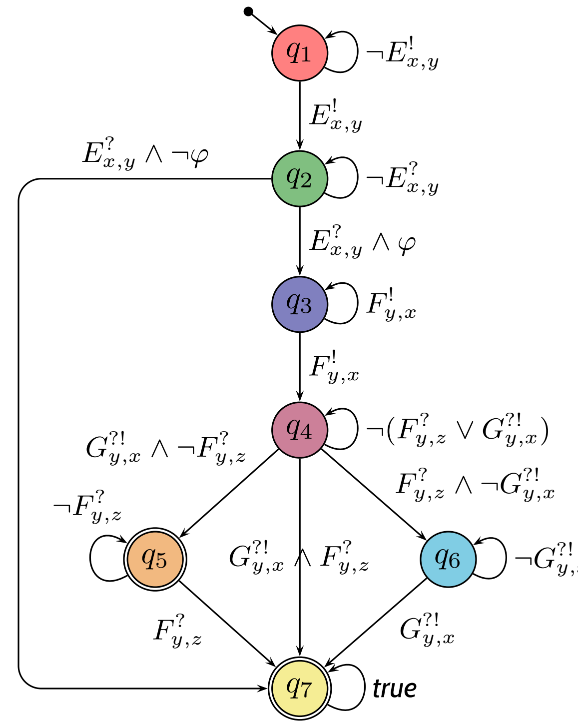
Successor Cut Example

$C \cap F = \emptyset$ – $C \cup F$ is a cut – only direct $\prec$-successors – same instance line on front pairwise unordered – sending of asynchronous reception already in



Language of LSC Body: Example





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

- Q is **the set of cuts** of $\mathscr{L}$, q_{ini} is the **instance heads** cut,
- $Expr_\mathcal{B}(X) = Expr_\mathscr{S}(\mathscr{E}, X)$ (for considered signature $\mathscr{S}$),
- $\rightarrow$ consists of **loops**, **progress transitions** (by $\rightsquigarrow_F$), and **legal exits** (cold cond./local inv.),
- $Q_F = \{C \in Q \mid \Theta(C) = \text{cold} \vee C = L\}$ is the set of cold cuts and the maximal cut.

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:

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

  where $expr : Bool \in Expr_{\mathscr{S}}$, $E \in \mathscr{E}$, $x, y \in X$ (or keyword env).

- We use

$$\mathscr{E}_{!?}(X) := \{E^!_{x,y}, E^?_{x,y} \mid E \in \mathscr{E}, x, y \in X\}$$

  to denote the set of **event expressions** over $\mathscr{E}$ and X.

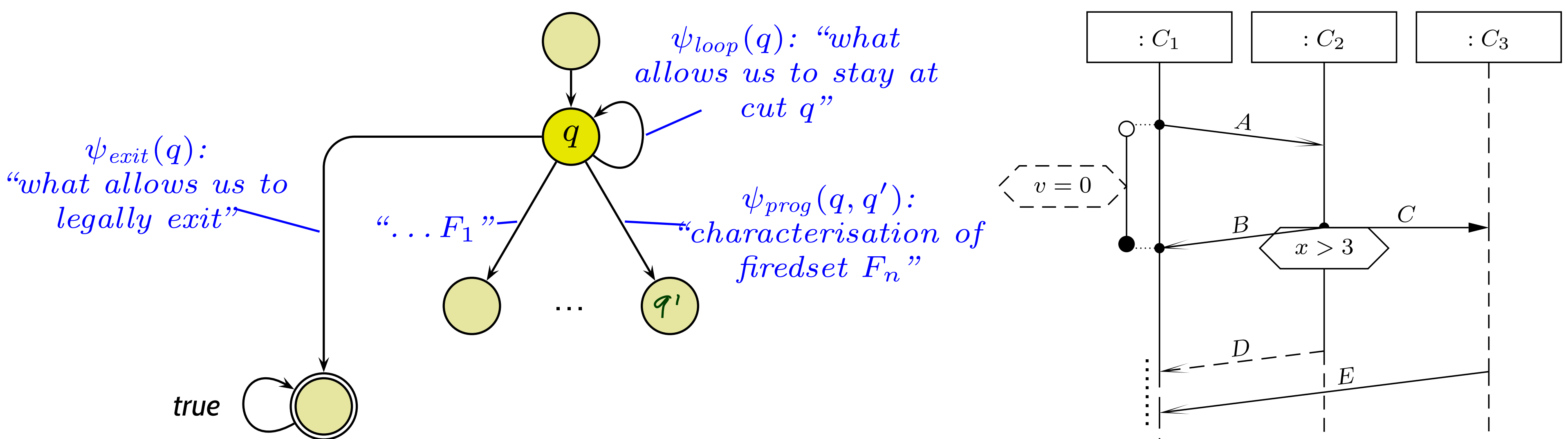
TBA Construction Principle

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

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

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

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



TBA Construction Principle

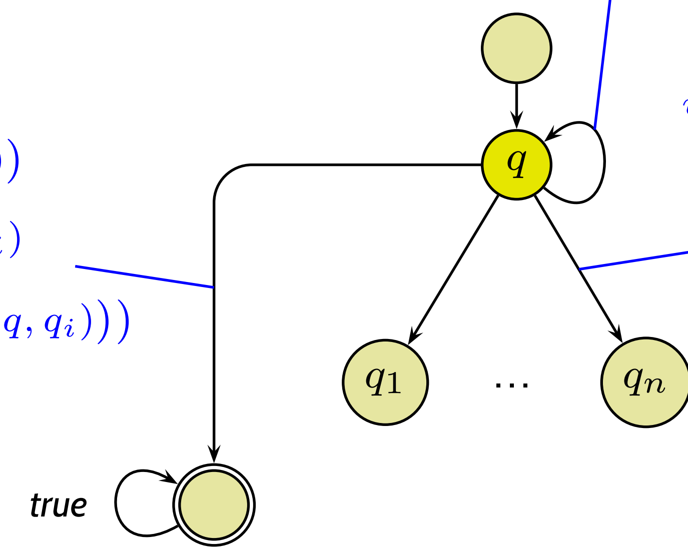
"Only" construct the transitions' labels:

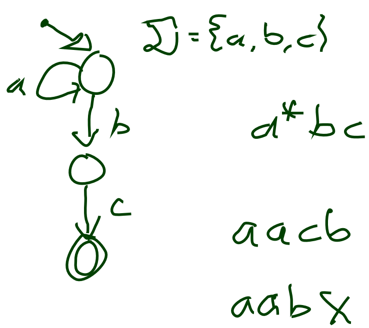
$$\rightarrow = \{(q, \psi_{loop}(q), q) \mid q \in Q\} \cup \{(q, \psi_{prog}(q, q'), q') \mid q \rightsquigarrow_F q'\} \cup \{(q, \psi_{exit}(q), L) \mid q \in Q\}$$

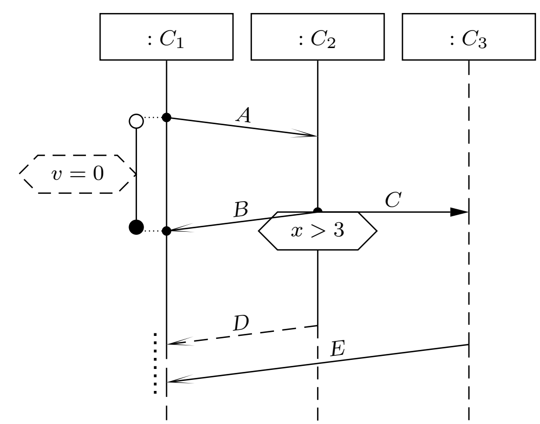
$$\psi_{loop}(q) = \underbrace{\psi^{\mathsf{Msg}}(q) \wedge \psi^{\mathsf{LocInv}}_{\mathsf{hot}}(q)}_{=:\psi^{\mathsf{hot}}_{loop}(q)} \wedge \psi^{\mathsf{LocInv}}_{\mathsf{cold}}(q)$$

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

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







Loop Condition

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

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

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

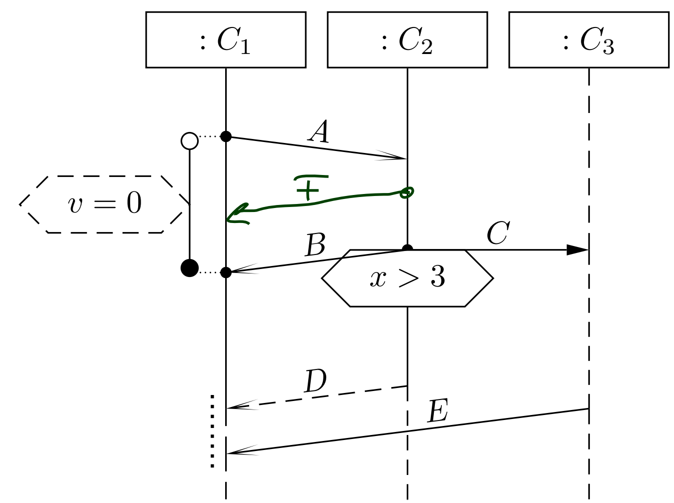
  A location l is called **front location** of cut C if and only if $\nexists\, l' \in L \bullet l \prec l'$.

  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 $l_1 \in q \wedge \iota_1 = \bullet$.~~)

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

- $x_l \in X$ is the logical variable associated with the instance line I which includes l, i.e. $l \in I$.

- $\mathsf{Msg}(F_1, \ldots, F_n) = \bigcup_{1 \leq i \leq n} \mathsf{Msg}(F_i)$



Progress Condition

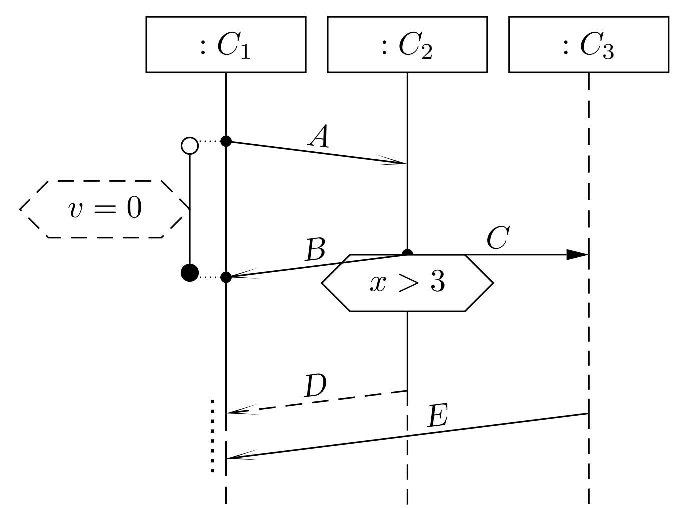
$$\psi^{\text{hot}}_{prog}(q, q_i) = \psi^{\text{Msg}}(q, q_n) \wedge \psi^{\text{Cond}}_{\text{hot}}(q, q_n) \wedge \psi^{\text{LocInv},\bullet}_{\text{hot}}(q_n)$$

- $$\psi^{\text{Msg}}(q, q_i) = \bigwedge_{\psi \in \text{Msg}(q_i \setminus q)} \psi \wedge \bigwedge_{j \neq i} \bigwedge_{\psi \in \text{Msg}(q_j \setminus q) \setminus \text{Msg}(q_i \setminus q)} \neg\psi \wedge \underbrace{\left(strict \implies \bigwedge_{\psi \in \text{Msg}(L) \setminus \text{Msg}(F_i)} \neg\psi\right)}_{=: \psi_{\text{strict}}(q, q_i)}$$
- $$\psi^{\text{Cond}}_{\theta}(q, q_i) = \bigwedge_{\gamma = (L, \phi) \in \text{Cond},\ \Theta(\gamma) = \theta,\ L \cap (q_i \setminus q) \neq \emptyset} \phi$$

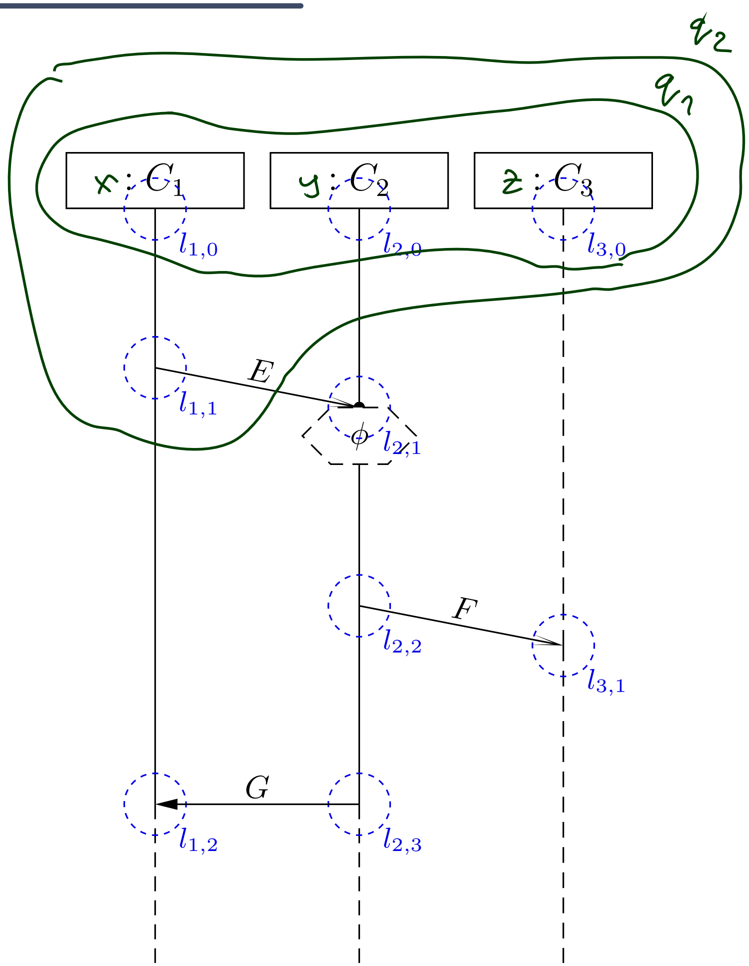
  fired-set
- $$\psi^{\text{LocInv},\bullet}_{\theta}(q, q_i) = \bigwedge_{\lambda = (l, \iota, \phi, l', \iota') \in \text{LocInv},\ \Theta(\lambda) = \theta,\ \lambda\ \bullet\text{-active at } q_i} \phi$$

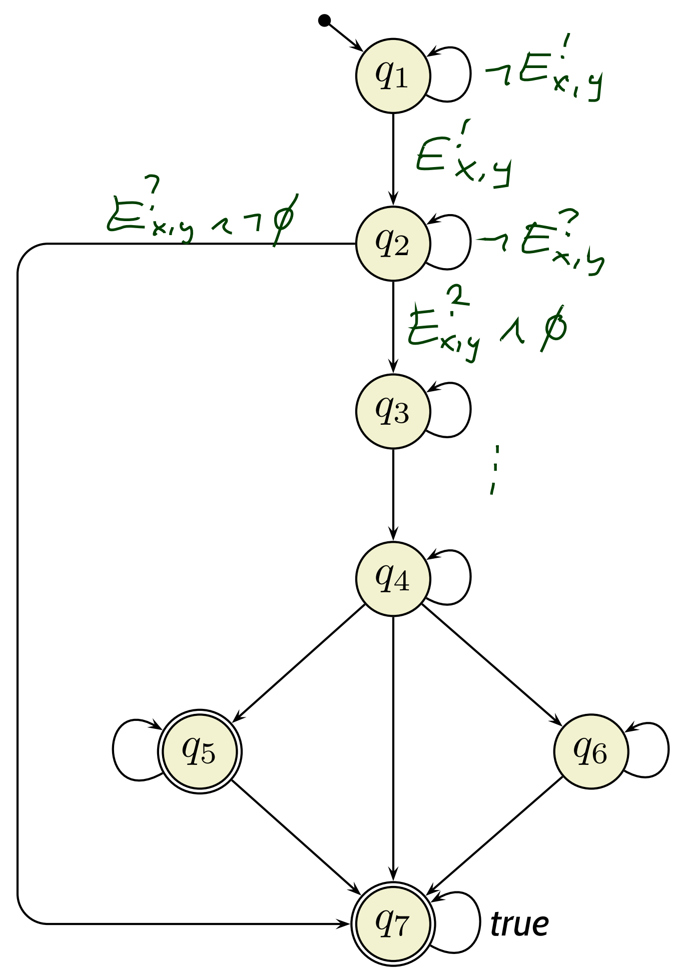
  Local invariant $(l_0, \iota_0, \phi, l_1, \iota_1)$ is **$\bullet$-active** at q if and only if
  - $l_0 \prec l \prec l_1$, or
  - $l = l_0 \wedge \iota_0 = \bullet$, or
  - $l = l_1 \wedge \iota_1 = \bullet$

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



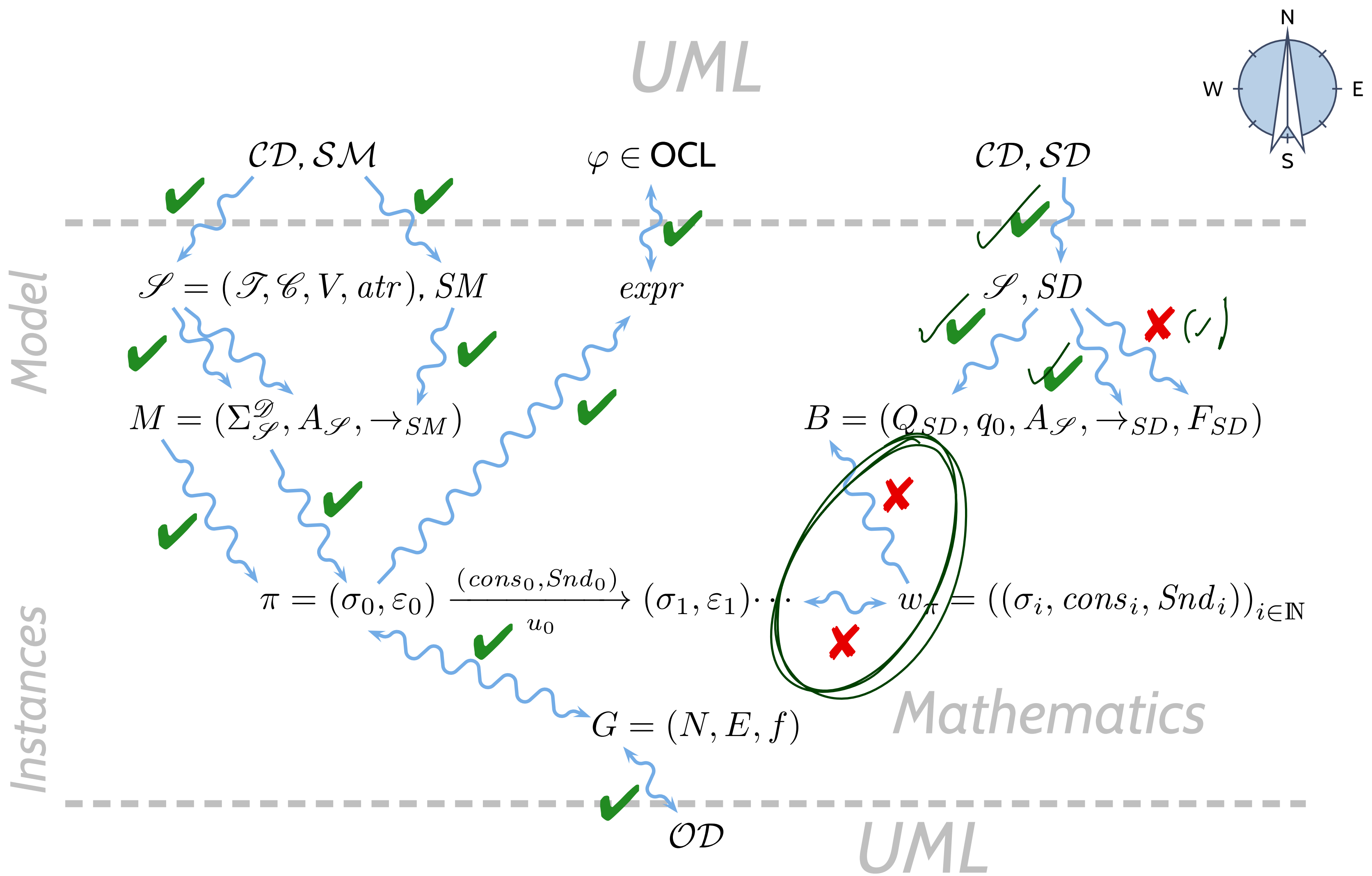
Example





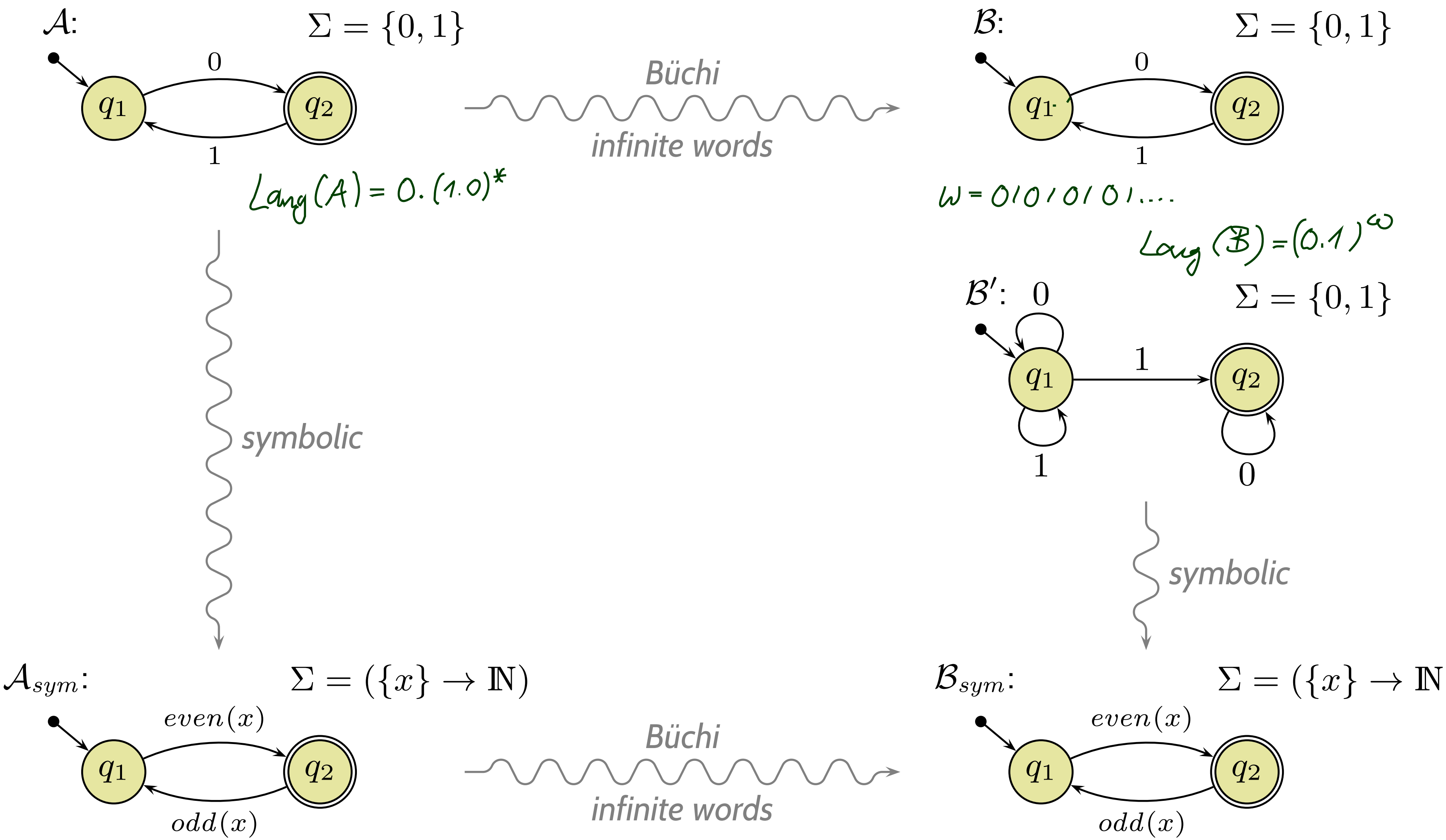
Using logical variables x, y, z for the instances lines (from left to right).

Course Map


UML
$\mathcal{CD}, \mathcal{SM}$
$\varphi \in$ OCL
$\mathcal{CD}, \mathcal{SD}$
N
W
E
S
Model
$\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr), SM$
$expr$
$\mathscr{S}, SD$
$M = (\Sigma_{\mathscr{S}}^{\mathscr{D}}, A_{\mathscr{S}}, \rightarrow_{SM})$
$B = (Q_{SD}, q_0, A_{\mathscr{S}}, \rightarrow_{SD}, F_{SD})$
Instances
$\pi = (\sigma_0, \varepsilon_0) \xrightarrow[u_0]{(cons_0, Snd_0)} (\sigma_1, \varepsilon_1) \cdots$
$w_\pi = ((\sigma_i, cons_i, Snd_i))_{i \in \mathbb{N}}$
$G = (N, E, f)$
Mathematics
$\mathcal{OD}$
UML

Excursion: Büchi Automata

From Finite Automata to Symbolic Büchi Automata



Symbolic Büchi Automata

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

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

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.

Word

Definition. Let X be a set of logical variables and let $Expr_{\mathcal{B}}(X)$ be a set of Boolean expressions over X.

A set $(\Sigma, \cdot \models . \cdot)$ is called an **alphabet** for $Expr_{\mathcal{B}}(X)$ if and only if

- for each $\sigma \in \Sigma$,
  - for each expression $expr \in Expr_{\mathcal{B}}$, and
    - for each valuation $\beta : X \to \mathscr{D}(X)$ of logical variables,

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

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

An **infinite sequence**

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

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

Run of TBA over Word

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

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

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

$$\varrho = q_0, q_1, q_2, \ldots \in Q^{\omega}$$

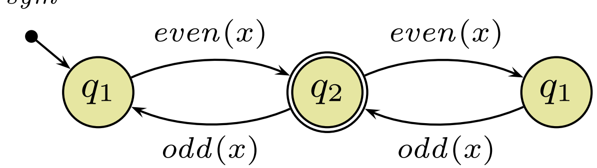
is called **run of $\mathcal{B}$ over w** under valuation $\beta : X \rightarrow \mathscr{D}(X)$ if and only if

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

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

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

Example: $\mathcal{B}_{sym}$: $\Sigma = (\{x\} \rightarrow \mathbb{N})$



The Language of a TBA

Definition.
We say TBA $\mathcal{B} = (Expr_{\mathcal{B}}(X), X, Q, q_{ini}, \rightarrow, Q_F)$ **accepts** the word

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

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

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

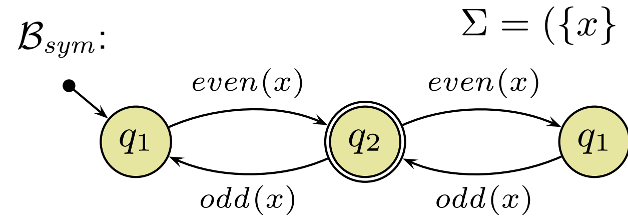
over w such that fair (or accepting) states are **visited infinitely often** by ϱ, i.e., such that

$$\forall\, i \in \mathbb{N}_0\ \exists\, j > i : q_j \in Q_F.$$

We call the set $\mathcal{L}(\mathcal{B}) \subseteq (Expr_{\mathcal{B}} \rightarrow \mathbb{B})^\omega$ of words that are accepted by $\mathcal{B}$ the **language of** $\mathcal{B}$.

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

Example:



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.