Softwaretechnik / Software-Engineering

Lecture 09: Live Sequence Charts

2015-06-11

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

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:

- Scenarios and Anti-Scenarios
- User Stories, Use Cases, Use Case Diagrams
- LSC: abstract and concrete syntax

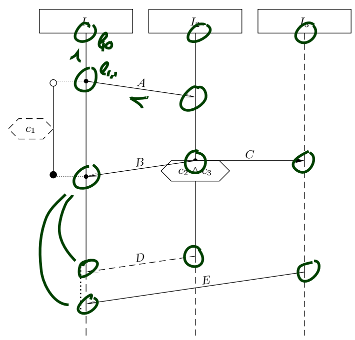
This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - Which are the cuts and firedsets of this LSC?
  - Construct the TBA of a given LSC body.
  - Given a set of LSCs, which scenario/anti-scenario/requirement is formalised by them?
  - Formalise this positive scenario/anti-scenario/requirement using LSCs.
- **Content:**
  - Excursion: automata accepting infinite words
  - Cuts and Firedsets, automaton construction
  - existential LSCs, pre-charts, universal LSCs
  - Requirements Engineering: conclusions

Recall: LSC Body Syntax

LSC Body Example

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



– 08 – 2015-06-08 – Slscsyn –

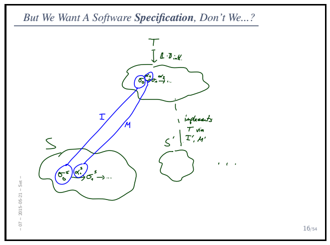
– 09 – 2015-06-11 – main –

LSC Semantics

– 09 – 2015-06-11 – main –

The Big Picture

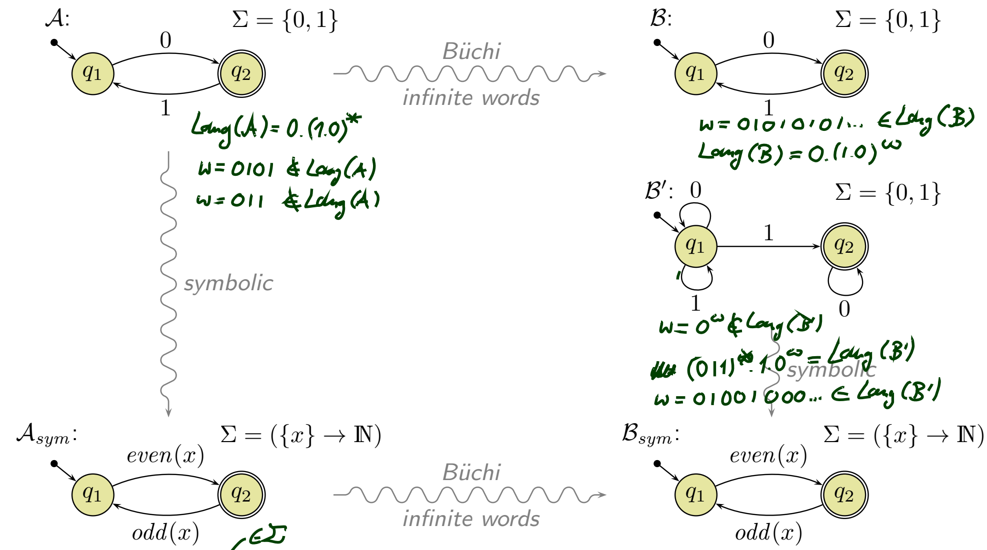
- **Recall**: **decision tables**
- By the standard semantics, a decision table T is **software**, $[\![T]\!] = \{\sigma_0 \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \cdots \mid \cdots\}$ is a set of computation paths.
- **Recall**: Decision tables as software specification:



- We want **the same** for LSCs.
- We will give a **procedure** to construct for each LSC $\mathscr{L}$ an **automaton** $\mathcal{B}(\mathscr{L})$. The language (or semantics) of $\mathscr{L}$ is the set of comp. paths **accepted** by $\mathcal{B}(\mathscr{L})$. Thus an LSC is also software.
- **Problem**: computation paths may be infinite $\rightarrow$ Büchi acceptance.

Excursion: Symbolic Büchi Automata

From Finite Automata to Symbolic Büchi Automata



∈Σ

w = (x=2)(x=3)(x=4) ∈ Lang(A_{sym})

w = (x=2)(x=3)(x=27) ∉ Lang(A_{sym})

Lang(A_{sym}) = { $x_0, \ldots, x_n$ | ∀i ≤ No even(x_i) ∧ odd(x_{i+1}) }

Symbolic Büchi Automata

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

$$\mathcal{B} = (\mathcal{C}, Q, q_{ini}, \to, Q_F)$$

where

- $\mathcal{C}$ is a set of atomic propositions,
- Q is a finite set of **states**,
- $q_{ini} \in Q$ is the initial state,
- $\to \subseteq Q \times \Phi(\mathcal{C}) \times Q$ is the finite **transition relation**.
  Each transitions $(q, \psi, q') \in \to$ from state q to state q' is labelled with a formula $\psi \in \Phi(\mathcal{C})$.
- $Q_F \subseteq Q$ is the set of **fair** (or accepting) states.

Run of TBA

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

$$w = \sigma_1, \sigma_2, \sigma_3, \dots \in (\mathcal{C} \rightarrow \mathbb{B})^\omega$$

an infinite word, each letter is a valuation of $\mathcal{C}_\mathcal{B}$.

An infinite sequence

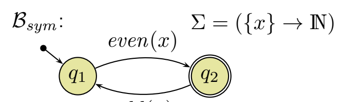
$$\varrho = q_0, q_1, q_2, \ldots \in Q^\omega$$

of states is called **run** of $\mathcal{B}$ over w if and only if

- $q_0 = q_{ini}$,
- for each $i \in \mathbb{N}_0$ there is a transition $(q_i, \psi_i, q_{i+1}) \in \rightarrow$ s.t. $\sigma_i \models \psi_i$.

$\omega = (x=0)(x=1)(x=4)(x=5)(x=8)(x=9)\ldots$

Example:



$R = q_1, q_2, q_1, q_2, q_1, q_2, \ldots$

– 09 – 2015-06-11 – Stba –

The Language of a TBA

Definition.
We say TBA $\mathcal{B} = (\mathcal{C}, Q, q_{ini}, \rightarrow, Q_F)$ **accepts** the word $w = (\sigma_i)_{i \in \mathbb{N}_0} \in (\mathcal{C} \rightarrow \mathbb{B})^\omega$ 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

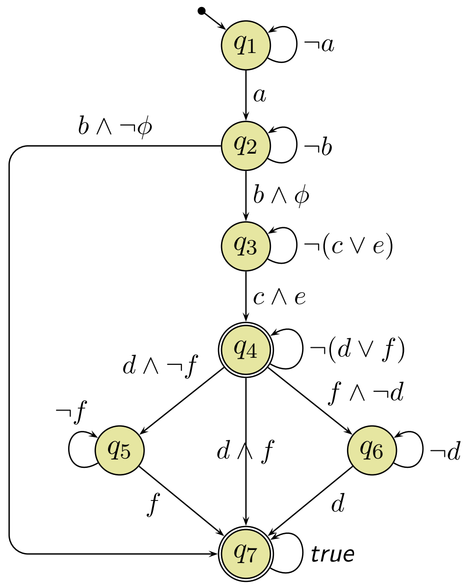
$$\forall\, i \in \mathbb{N}_0\ \exists\, j > i : q_j \in Q_F.$$

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

– 09 – 2015-06-11 – Stba –

Example

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



LSC Semantics: TBA Construction

LSC Semantics: It's in the Cuts!

Definition. Let $((\mathcal{L}, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta)$ be an LSC body.

A non-empty set $\emptyset \neq C \subseteq \mathcal{L}$ is called a **cut** of the LSC body iff C

- is **downward closed**, i.e.
$$\forall\, l, l' \in \mathcal{L} \bullet l' \in C \wedge l \preceq l' \implies l \in C,$$
- is **closed** under **simultaneity**, i.e.
$$\forall\, l, l' \in \mathcal{L} \bullet l' \in C \wedge l \sim l' \implies l \in C, \text{ and}$$
- comprises at least **one location per instance line**, i.e.
$$\forall\, I \in \mathcal{I} \bullet C \cap I \neq \emptyset.$$

The temperature function is extended to cuts as follows:

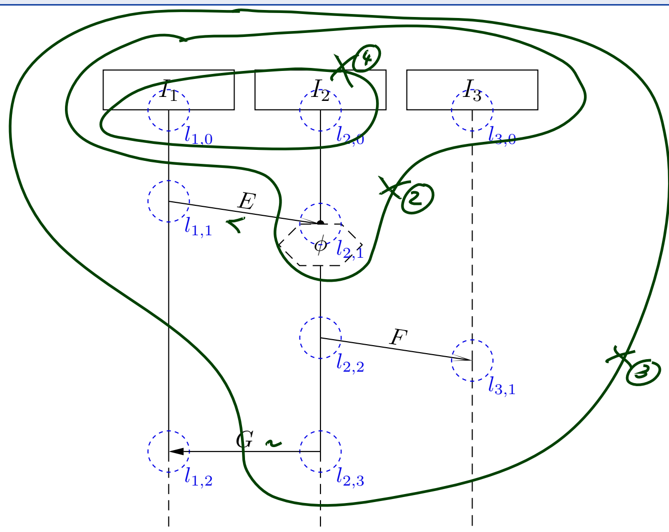
$$\Theta(C) = \begin{cases} \mathsf{hot} & \text{, if } \exists\, l \in C \bullet (\nexists\, l' \in C \bullet l \prec l') \wedge \Theta(l) = \mathsf{hot} \\ \mathsf{cold} & \text{, otherwise} \end{cases}$$

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

Cut Examples

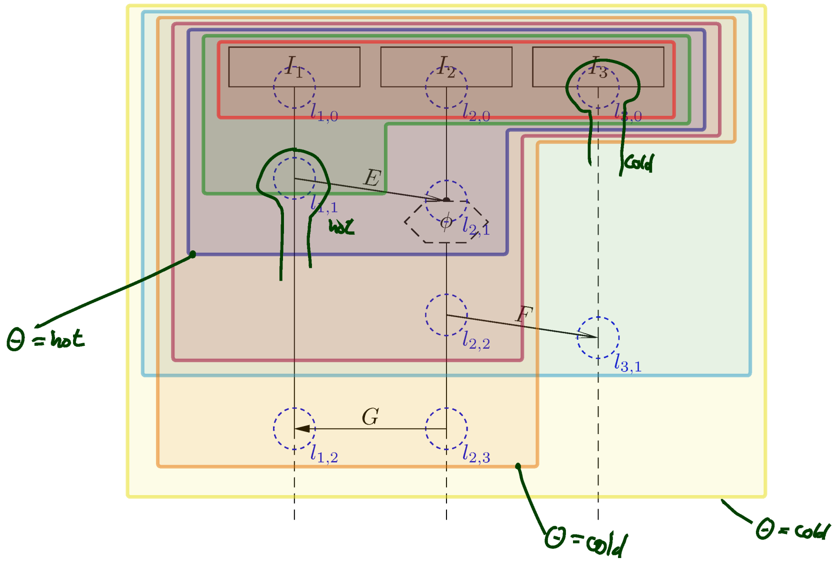
1 2 3 4

$\emptyset \neq C \subseteq \mathcal{L}$ — downward closed — simultaneity closed — at least one loc. per instance line



Cut Examples

$\emptyset \neq C \subseteq \mathcal{L}$ — 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 $\mathcal{L}$ as follows:

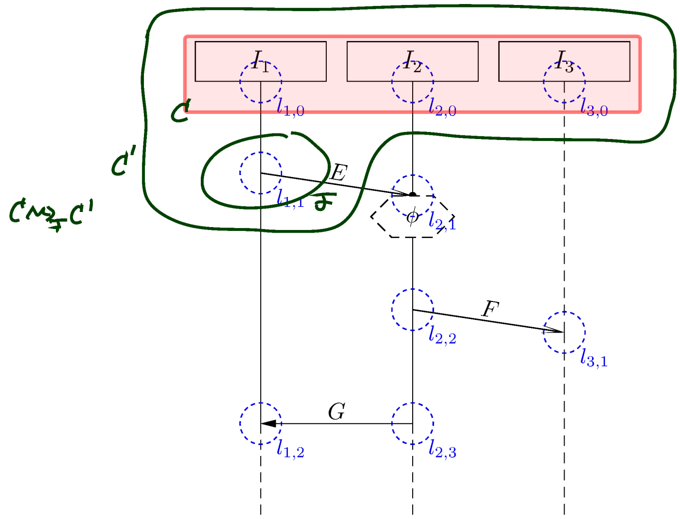
Definition.
Let $C \subseteq \mathcal{L}$ bet a cut of LSC body $((\mathcal{L}, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta)$.
A set $\emptyset \neq \mathcal{F} \subseteq \mathcal{L}$ is called fired-set $\mathcal{F}$ of C if and only if

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

The cut $C' = C \dot\cup \mathcal{F}$ is called direct successor of C via $\mathcal{F}$, denoted by $C \rightsquigarrow_{\mathcal{F}} C'$.

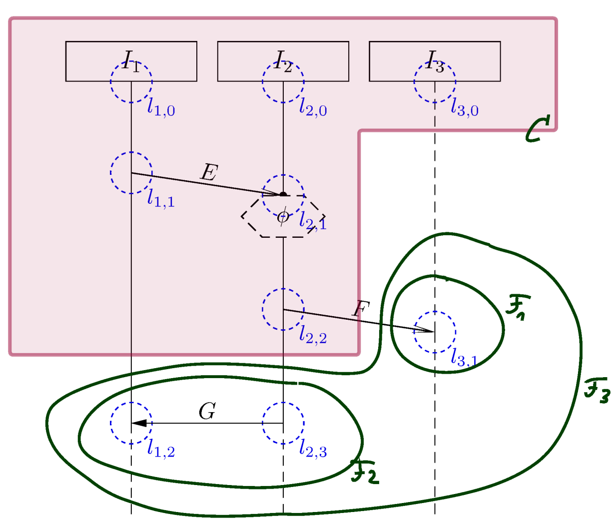
Successor Cut Example

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

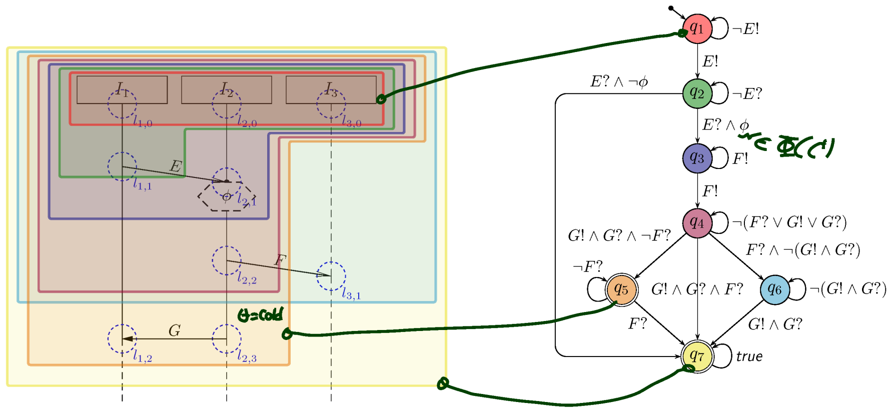


Successor Cut Example

$C \cap \mathcal{F} = \emptyset$ — $C \cup \mathcal{F}$ 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 C and $\mathcal{E}$ is $(\mathcal{C}, Q, q_{ini}, \rightarrow, Q_F)$ with

- Q is **the set of cuts** of $\mathscr{L}$, q_{ini} is the **instance heads** cut,
- $\mathcal{C} = C \cup \mathcal{E}_{!?}$, where $\mathcal{E}_{!?} = \{E!, E? \mid E \in \mathcal{E}\}$,
- $\rightarrow$ 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.

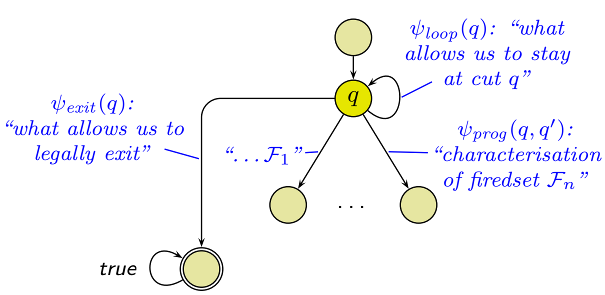
TBA Construction Principle

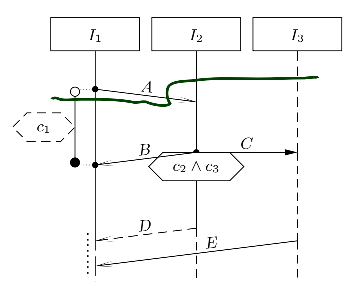
Recall: The TBA $\mathcal{B}(\mathscr{L})$ of LSC $\mathscr{L}$ is $(\mathcal{C}, Q, q_{ini}, \rightarrow, Q_F)$ with

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

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

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





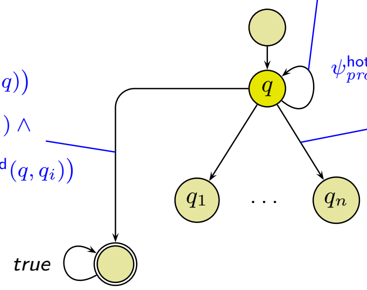
TBA Construction Principle

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

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

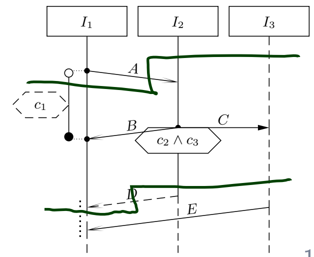
$$\psi_{loop}(q) = \underbrace{\psi^{\mathsf{Msg}}(q) \wedge \psi^{\mathsf{LocInv}}_{\mathsf{hot}}(q)}_{=:\psi^{\mathsf{hot}}_{loop}(q)} \wedge \psi^{\mathsf{LocInv}}_{\mathsf{cold}}(q)$$

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

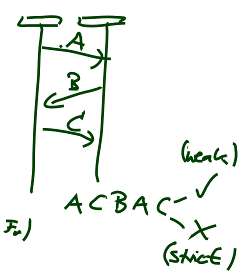


$\psi_{prog}(q, q_n) =$
$\psi^{\mathsf{hot}}_{prog}(q, q_n) \wedge \psi^{\mathsf{Cond}}_{\mathsf{cold}}(q, q_n) \wedge \psi^{\mathsf{LocInv},\bullet}_{\mathsf{cold}}(q, q_n)$

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



Loop Condition



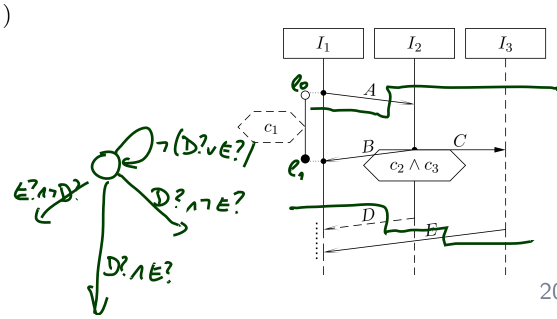
none of the fired-set msg. is observed

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

- $\psi^{\mathsf{Msg}}(q) = \neg \bigvee_{1 \le i \le n} \psi^{\mathsf{Msg}}(q, q_i) \wedge \big(strict \implies \bigwedge_{\psi \in \mathcal{E}_{!?} \cap \mathsf{Msg}(\mathcal{L})} \neg\psi\big)$ Msg($\mathcal{F}_1, \ldots, \mathcal{F}_n$)
- $\psi^{\mathsf{LocInv}}_{\theta}(q) = \bigwedge_{\ell = (l, \iota, \phi, l', \iota') \in \mathsf{LocInv},\ \Theta(\ell) = \theta,\ \ell \text{ active at } q} \phi$

  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 $(l_0, \iota_0, \phi, l_1, \iota_1)$ is **active** at cut (!) q if and only if $l_0 \preceq l \preceq l_1$ for some front location l of cut (!) q.
- $\mathsf{Msg}(\mathcal{F}) = \{E! \mid (l, E, l') \in \mathsf{Msg},\ l \in \mathcal{F}\} \cup \{E? \mid (l, E, l') \in \mathsf{Msg},\ l' \in \mathcal{F}\}$
- $\mathsf{Msg}(\mathcal{F}_1, \ldots, \mathcal{F}_n) = \bigcup_{1 \le i \le n} \mathsf{Msg}(\mathcal{F}_i)$



Progress Condition

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

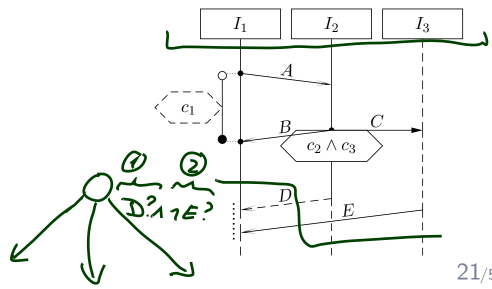
(1) the msgs. of firedset $\mathcal{F}_i$ — not msg. combination of any other firedset of q (2)

- $\psi^{\text{Msg}}(q, q_i) = \bigwedge_{\psi \in \text{Msg}(q_i \setminus q)} \psi \wedge \bigwedge_{j \neq i} \bigwedge_{\psi \in (\text{Msg}(q_j \setminus q) \setminus \text{Msg}(q_i \setminus q))} \neg\psi$
  $\wedge \left(strict \implies \bigwedge_{\psi \in (\mathcal{E}_{!?} \cap \text{Msg}(\mathcal{L})) \setminus \text{Msg}(\mathcal{F}_i)} \neg\psi \right)$
- $\psi^{\text{Cond}}_{\theta}(q, q_i) = \bigwedge_{\gamma = (L, \phi) \in \text{Cond},\ \Theta(\gamma) = \theta,\ L \cap (q_i \setminus q) \neq \emptyset} \phi$

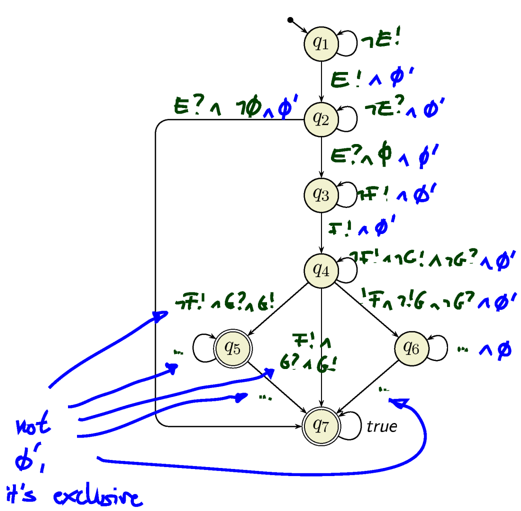
  condition has one location in the firedset $q_i \setminus q$
- $\psi^{\text{LocInv},\bullet}_{\theta}(q, q_i) = \bigwedge_{\lambda = (l, \iota, \phi, l', \iota') \in \text{LocInv},\ \Theta(\lambda) = \theta,\ \lambda\ \bullet\text{-active at } q_i} \phi$

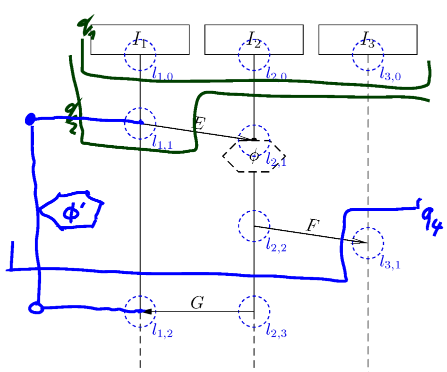
  Local invariant $(l_0, \iota_0, \phi, l_1, \iota_1)$ is **$\bullet$-active** at q if and only if
  - $l_0 \prec l \prec l_1$, or
  - $l = l_0 \wedge \iota_0 = \bullet$, or
  - $l = l_1 \wedge \iota_1 = \bullet$

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



Example



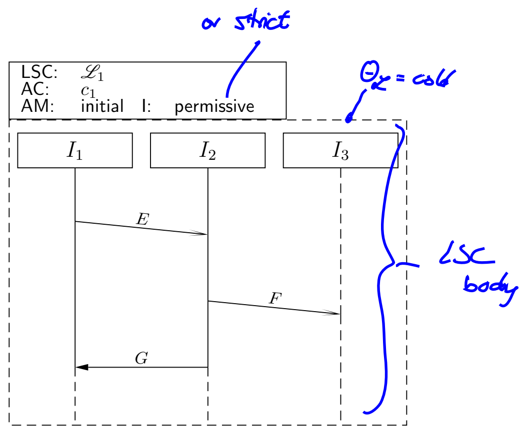


Finally: The LSC Semantics

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

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

Concrete syntax:



Finally: The LSC Semantics

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

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

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

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

where $ac = ac_0 \wedge \psi^{\mathsf{Cond}}_{\mathsf{cold}}(\emptyset, C_0) \wedge \psi^{\mathsf{Msg}}(\emptyset, C_0)$; C_0 is the minimal (or **instance heads**) cut.

References

References

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

ITU-T (2011). *ITU-T Recommendation Z.120: Message Sequence Chart (MSC)*, 5 edition.

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.