Theoretical Foundations of the UML

Lecture 13: Verifying PDL Formulas

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/i2/uml09100/

7. Januar 2013

RWTH AACHEN UNIVERSITY

Outline

RWTH AACHEN UNIVERSITY

Overview

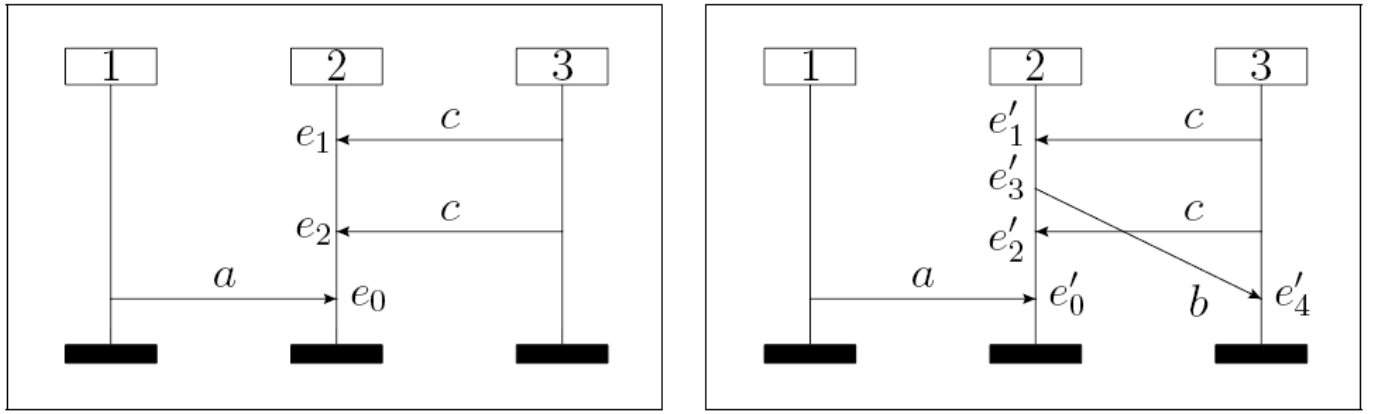
RWTH AACHEN UNIVERSITY

A logic for MSCs

- This lecture will be devoted to a logic that is interpreted over MSCs
- The logic is used to umambigously express properties of MSCs
  - does a given MSC M satisfy the logical formula φ?
- And to characterise a set of MSCs by means of a logical formula
  - all MSCs that satisfy the formula φ
- Our logic is a variant of propositional dynamic logic (PDL) [Fischer & Ladner, 1979]
  - combines easy-to-grasp concepts such as regular expressions and Boolean operators
- We consider syntax, semantics, examples and the membership problem.

RWTH AACHEN UNIVERSITY

Some informal examples



1. The (unique) maximal event of M is labeled by $?(2, 1, a)$ Yes. No.
2. The maximal event on process 2 is labeled by $?(2, 1, a)$ Yes. Yes.
3. No two consecutive events are labeled with $?(2, 3, c)$ No. Yes.
4. The number of send events at process 3 is odd. No. No.

RWTH AACHEN UNIVERSITY

Overview

RWTH AACHEN UNIVERSITY

Local formulas

Local formulas

These are statements over single events in an MSC. That is, an event either satisfies or refutes such formula.

Example local formulas

- $!(1,2,a)$ The current event is labeled with $!(1,2,a)$
- $\langle \mathsf{proc} \rangle true$ There is a next event at the same process
- $\langle \mathsf{proc}; \mathsf{proc} \rangle true$ There are (at least) two next events at this process
- $[\mathsf{proc}]^{-1} false$ There is no preceding event at this process
- $\langle \mathsf{msg} \rangle true$ This send event matches a (next) receive event
- $\langle \{\, !(1,2,a) \,\}; \mathsf{proc} \rangle\, ?(1,2,b)$ Event $!(1,2,a)$ is followed by $?(1,2,b)$
- $[\mathsf{proc} + \mathsf{msg}] \neg \{\, !(1,2,a) \,\}$ Any next event differs from $!(1,2,a)$

RWTH AACHEN UNIVERSITY

Local formulas

Definition (Syntax of local formulas)

For communication action $\sigma \in Act$ and path expression α, the grammar of local formulas is given by:

$$\varphi \ ::= \ true \ \mid \ \sigma \ \mid \ \neg\varphi \ \mid \ \varphi \vee \varphi \ \mid \ \langle\alpha\rangle\varphi \ \mid \ \langle\alpha\rangle^{-1}\varphi$$

Path expressions will be defined later on.

Definition (Derived operators)

$$\begin{aligned} false &= \neg true \\ \varphi_1 \wedge \varphi_2 &= \neg(\neg\varphi_1 \vee \neg\varphi_2) \\ \varphi_1 \rightarrow \varphi_2 &= \neg\varphi_1 \vee \varphi_2 \\ [\alpha]\varphi &= \neg\langle\alpha\rangle\neg\varphi \\ [\alpha]^{-1}\varphi &= \neg\langle\alpha\rangle^{-1}\neg\varphi \end{aligned}$$

Intuitive meaning of local formulas

$true$	Valid statement. Satisfied by every event.
σ	Current event is labelled with σ
$\neg\varphi$	Current event does not satisfy φ
$\varphi_1 \vee \varphi_2$	Current event satisfies φ_1 or φ_2
$\langle\alpha\rangle\varphi$	Some forward path satisfying α reaches an event satisfying φ
$\langle\alpha\rangle^{-1}\varphi$	Some backward path α reaches an event satisfying φ
$[\alpha]\varphi$	All forward paths satisfying α reach an event satisfying φ
$[\alpha]^{-1}\varphi$	All backward paths satisfying α reach an event satisfying φ

How are path expressions like α defined?

RWTH AACHEN UNIVERSITY

Path expressions

Definition (Syntax of local formulas)

For communication action $\sigma \in Act$ and path expression α, the grammar of local formulas is given by:

$$\varphi \ ::= \ true \ \mid \ \sigma \ \mid \ \neg\varphi \ \mid \ \varphi \vee \varphi \ \mid \ \langle\alpha\rangle\varphi \ \mid \ \langle\alpha\rangle^{-1}\varphi$$

Definition (Syntax of path expressions)

For local formula φ, the grammar of path expressions is given by:

$$\alpha \ ::= \ \{\varphi\} \ \mid \ \mathsf{proc} \ \mid \ \mathsf{msg} \ \mid \ \alpha;\alpha \ \mid \ \alpha + \alpha \ \mid \ \alpha^*$$

RWTH AACHEN UNIVERSITY

Intuitive meaning of path expressions

- $\{\,\varphi\,\}$ specifies an event that satisfies φ
- proc requires a (direct) successor relation between events at the same process
- msg requires a matching between current event and a receive event
- The composition $\alpha;\beta$ defines the set of pairs (e, e') for which there exist event e'' such that $(e, e'') \models \alpha$ and $(e'', e') \models \beta$
- $\alpha + \beta$ denotes the union of the relations α and β
- α^* denotes the reflexive and transitive closure of the relation α

RWTH AACHEN UNIVERSITY

Intuitive meaning of local formulas

- Local formulas are interpreted over MSC events
- Event e satisfies $\underbrace{!(p,q,a)}_{\sigma}$ iff e is labelled with action $\underbrace{!(p,q,a)}_{\sigma}$
- Path expression α defines a binary relation between events:
  1. $\{\varphi\}$ is the set of pairs (e,e') such that e satisfies φ
  2. $(e,e') \models \mathsf{proc}$ iff e and e' reside at the same process and e' is a direct successor of e wrt. $<_p$
  3. $(e,e') \models \mathsf{msg}$ iff e' is the matching event of e, i.e., $e' = m(e)$

RWTH AACHEN UNIVERSITY

Forward and backward local formulas

- Event e satisfies $\langle\alpha\rangle\varphi$ iff there is an event e' such that a path from e to e' satisfies α and e' satisfies φ

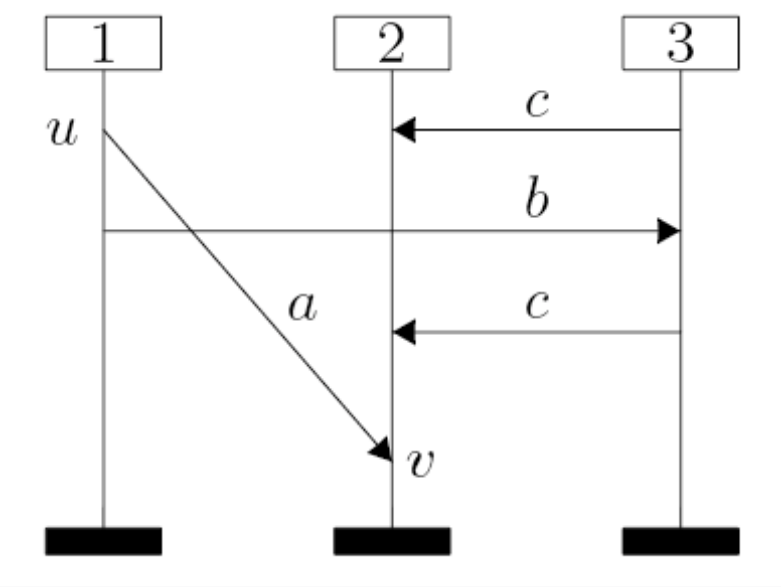
  Formula $\langle\alpha\rangle\varphi$ looks "forward" along the partial order of the MSC starting from the current event

- The interpretation of $\langle\alpha\rangle^{-1}\varphi$ is dual, i.e., e satisfies it iff there is an event e' such that some path from e' to e satisfies α and e' satisfies φ

  Formula $\langle\alpha\rangle^{-1}\varphi$ looks "backward" along the partial order of the MSC starting from the current event

RWTH AACHEN UNIVERSITY

Example



1. $u \models !(1, 2, a)$ — u is labelled with the action $!(1, 2, a)$
2. $u \models [\mathsf{proc}]^{-1}\, \mathit{false}$ — u is the first event on the process line
3. $u \models \langle(\mathsf{proc} + \mathsf{msg})^*\rangle ?(2, 1, a)$ — event u happens before the event v

RWTH AACHEN UNIVERSITY

Semantics of local formulas (1)

Definition (Syntax of local formulas)

For communication action $\sigma \in Act$ and path expression α:

$$\varphi \ ::= \ true \ \mid \ \sigma \ \mid \ \neg\varphi \ \mid \ \varphi \vee \varphi \ \mid \ \langle\alpha\rangle\varphi \ \mid \ \langle\alpha\rangle^{-1}\varphi$$

Definition (Semantics of base local formulas)

Let $M = (\mathcal{P}, E, \mathcal{C}, l, m, <) \in \mathbb{M}$ be an MSC and $e \in E$.

Binary relation $\models$ is defined such that $((M, e), \varphi) \in \models$ iff event e of MSC M satisfies local formula φ. We write $M, e \models \varphi$ for $((M, e), \varphi) \in \models$.

$$\begin{aligned} M, e \models true & \qquad \text{for all } e \in E \\ M, e \models \sigma \quad \text{iff} & \quad l(e) = \sigma \\ M, e \models \neg\varphi \quad \text{iff} & \quad \text{not } M, e \models \varphi \\ M, e \models \varphi_1 \vee \varphi_2 \quad \text{iff} & \quad M, e \models \varphi_1 \text{ or } M, e \models \varphi_2 \end{aligned}$$

Semantics of local formulas (2)

Definition (Semantics of forward path formulas)

Let $M = (\mathcal{P}, E, \mathcal{C}, l, m, <) \in \mathbb{M}$ be an MSC and $e \in E$.

$$
\begin{aligned}
e \models \langle \{\psi\} \rangle \varphi \quad &\text{iff} \quad e \models \psi \text{ and } e \models \varphi \\
e \models \langle \mathsf{proc} \rangle \varphi \quad &\text{iff} \quad \exists e' \in E.\, e \lessdot_p e' \text{ and } e' \models \varphi \\
e \models \langle \mathsf{msg} \rangle \varphi \quad &\text{iff} \quad \exists e' \in E.\, e' = m(e) \text{ and } e' \models \varphi \\
e \models \langle \alpha_1 ; \alpha_2 \rangle \varphi \quad &\text{iff} \quad e \models \langle \alpha_1 \rangle \langle \alpha_2 \rangle \varphi \\
e \models \langle \alpha_1 + \alpha_2 \rangle \varphi \quad &\text{iff} \quad e \models \langle \alpha_1 \rangle \varphi \text{ or } e \models \langle \alpha_2 \rangle \varphi \\
e \models \langle \alpha^* \rangle \varphi \quad &\text{iff} \quad \exists n \in \mathbb{N}.\, e \models (\langle \alpha \rangle)^n \varphi
\end{aligned}
$$

Where $e \lessdot_p e'$ iff $e <_p e'$ and $\neg(\exists e''.\, e <_p e'' <_p e')$, i.e., e' is a direct successor of e under $<_p$.

Semantics of local formulas (3)

Definition (Semantics of backward path formulas)

Let $M = (\mathcal{P}, E, \mathcal{C}, l, m, <) \in \mathbb{M}$ be an MSC and $e \in E$.

$$
\begin{aligned}
e \models \langle \{\psi\} \rangle^{-1} \varphi \quad &\text{iff} \quad e \models \psi \text{ and } e \models \varphi \\
e \models \langle \mathsf{proc} \rangle^{-1} \varphi \quad &\text{iff} \quad \exists e' \in E.\, e' <_p e \text{ and } e' \models \varphi \\
e \models \langle \mathsf{msg} \rangle^{-1} \varphi \quad &\text{iff} \quad \exists e' \in E.\, e' = m^{-1}(e) \text{ and } e' \models \varphi \\
e \models \langle \alpha_1 ; \alpha_2 \rangle^{-1} \varphi \quad &\text{iff} \quad e \models \langle \alpha_1 \rangle^{-1} \langle \alpha_2 \rangle^{-1} \varphi \\
e \models \langle \alpha_1 + \alpha_2 \rangle^{-1} \varphi \quad &\text{iff} \quad e \models \langle \alpha_1 \rangle^{-1} \varphi \text{ or } e \models \langle \alpha_2 \rangle^{-1} \varphi \\
e \models \langle \alpha^* \rangle^{-1} \varphi \quad &\text{iff} \quad \exists n \in \mathbb{N}.\, e \models \left(\langle \alpha \rangle^{-1} \right)^n \varphi
\end{aligned}
$$

RWTH AACHEN UNIVERSITY

Overview

RWTH AACHEN UNIVERSITY

PDL formulas

Definition (Syntax of PDL formulas)

For local formula φ, the grammar of PDL formulas is given by:

$$\Phi \ ::= \ \exists\varphi \ \mid \ \forall\varphi \ \mid \ \Phi \wedge \Phi \ \mid \ \Phi \vee \Phi$$

RWTH AACHEN UNIVERSITY

Intuitive meaning of PDL formulas

- MSC M satisfies $\exists\varphi$ if it has some event e satisfying φ
- MSC M satisfies $\exists\langle\alpha\rangle\varphi$ if from some event e in M, there exists an α-labelled path from e to an event e', say, satisfying φ
- MSC M satisfies $\exists[\alpha]\varphi$ if from some event e in M, any event that can be reached via an α-labelled path satisfies φ

RWTH AACHEN UNIVERSITY

Semantics of PDL formulas

Definition (Semantics of PDL formulas)

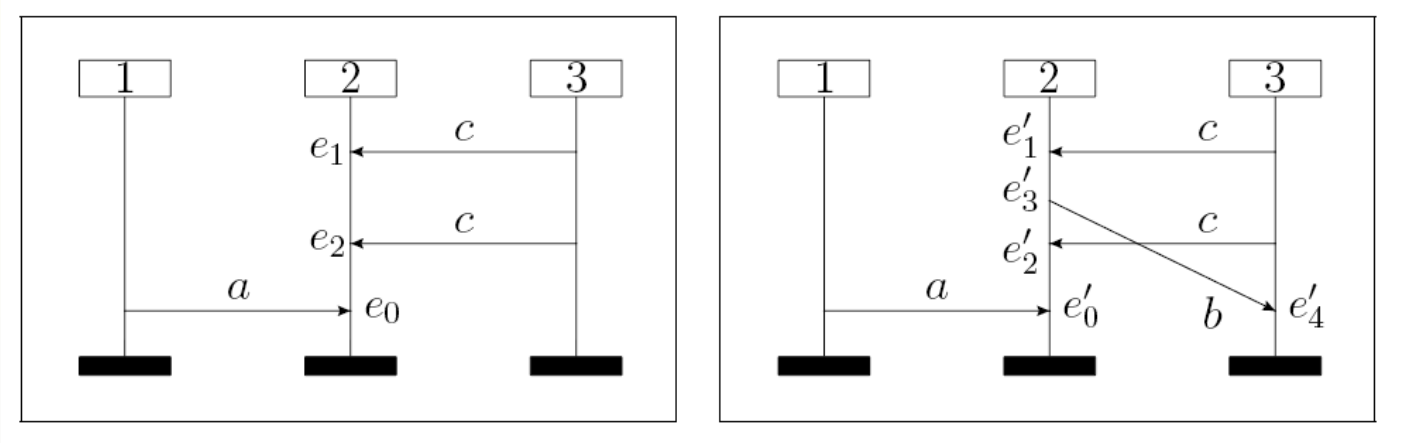
Let $M = (\mathcal{P}, E, \mathcal{C}, l, m, <) \in \mathbb{M}$ be an MSC.

$(M, \Phi) \in \models$ iff PDL formula Φ holds in MSC M.

$$
\begin{aligned}
M \models \exists\varphi \quad &\text{iff} \quad \exists e \in E.\, M, e \models \varphi \\
M \models \forall\varphi \quad &\text{iff} \quad \forall e \in E.\, M, e \models \varphi \\
M \models \Phi_1 \wedge \Phi_2 \quad &\text{iff} \quad M \models \Phi_1 \text{ and } M \models \Phi_2 \\
M \models \Phi_1 \vee \Phi_2 \quad &\text{iff} \quad M \models \Phi_1 \text{ or } M \models \Phi_2
\end{aligned}
$$

RWTH AACHEN UNIVERSITY

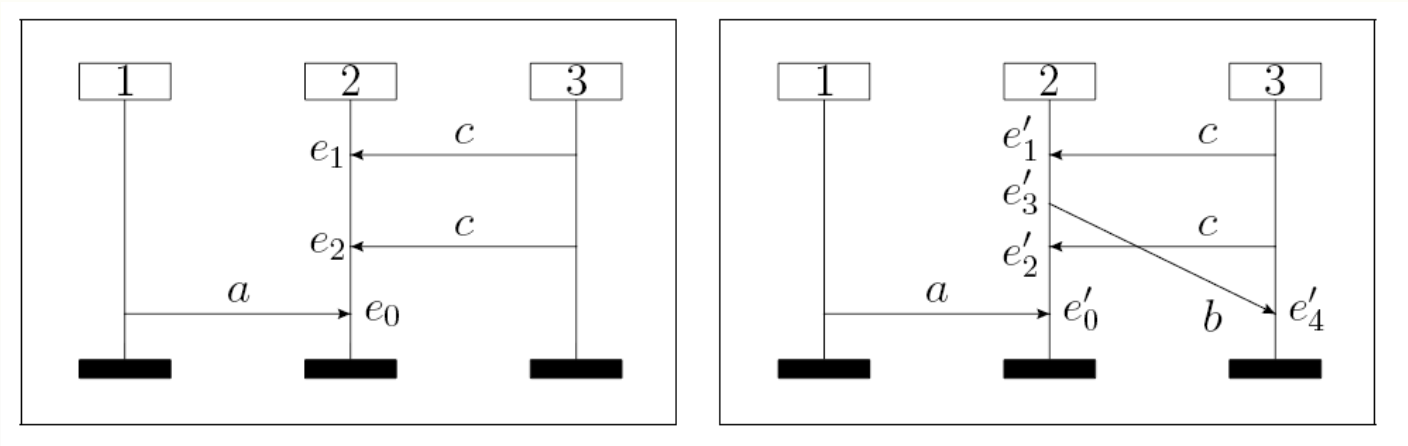
Example (1)



- The (unique) maximal event of M is labelled by $?(2, 1, a)$ Yes. No.
- $\forall\,(\langle(\mathsf{proc} + \mathsf{msg})^*\rangle([\mathsf{proc}]\,\mathit{false} \wedge ?(2, 1, a)))$ Yes. No.

RWTH AACHEN UNIVERSITY

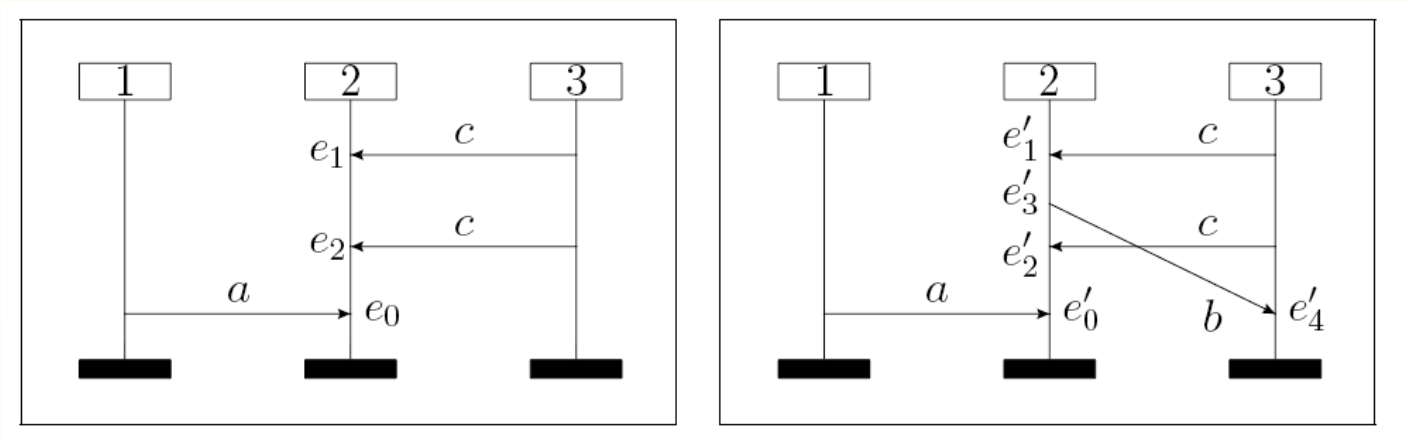
Example (2)



- The maximal event on process 2 is labelled by $?(2, 1, a)$ Yes. Yes.
- $\exists\,([\mathsf{proc}]\,\mathit{false} \wedge ?(2, 1, a))$ Yes. Yes.

RWTH AACHEN UNIVERSITY

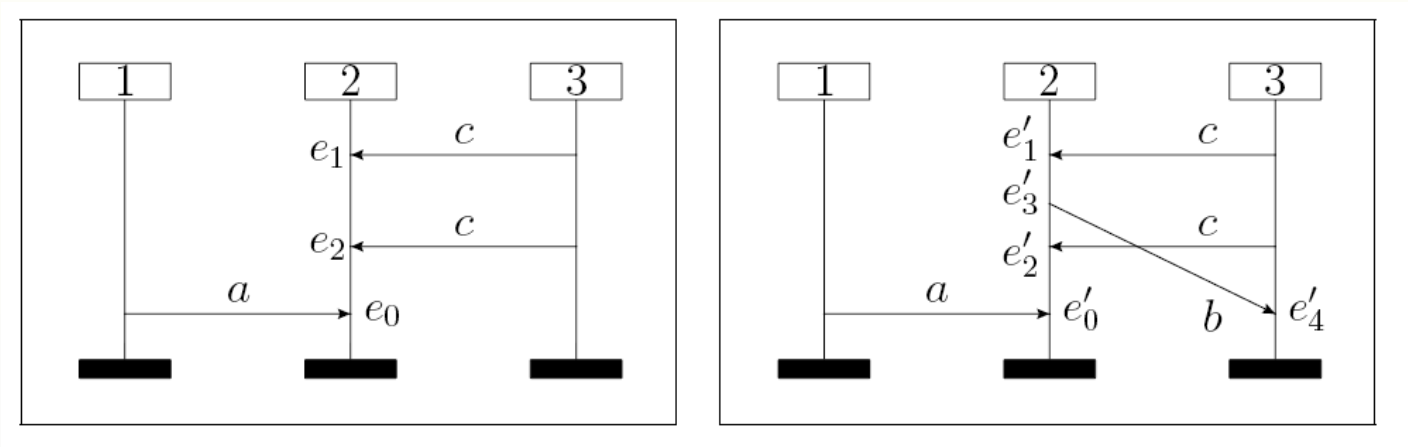
Example (3)



- No two consecutive events are labelled with $?(2, 3, c)$ No. Yes.
- $\forall\,([?(2, 3, c); \mathsf{proc}; ?(2, 3, c)]\, \mathit{false})$ No. Yes.

RWTH AACHEN UNIVERSITY

Example (4)



- The number of send events at process 3 is odd. No. No.
- See next slide

Example (slightly changed)

MSC M has an even number of messages sent from process 1 to 2:

$$\forall\Big(\quad \underbrace{[\mathsf{proc}]^{-1}\, \mathit{false} \wedge P_1}_{\text{minimal event on process 1}} \quad \rightarrow \quad \langle\alpha\rangle \quad \underbrace{[\mathsf{proc}]\, \mathit{false}}_{\text{maximal event on process}} \quad \Big)$$

where $P_1 = \bigvee_{j \in \mathcal{P}, j \neq 1} (!_{1,j} \vee ?_{1,j})$ with $!_{1,j} = \bigvee_{a \in \mathcal{C}} !(1, j, a)$ and $?_{1,j}$ is defined in a similar way, i.e., $e \models P_1$ iff e occurs at process 1.

Path expression α is defined by:

$$\alpha \;=\; ((\{\neg !_1\}; \mathsf{proc})^*; \{!_1\}; \mathsf{proc}; (\{\neg !_1\}; \mathsf{proc})^*; \{!_1\}; \mathsf{proc}; (\{\neg !_1\}; \mathsf{proc})^*)^*$$

and where $!_1$ abbreviates $\bigvee_{a \in \mathcal{C}} !(1, 2, a)$

RWTH AACHEN UNIVERSITY

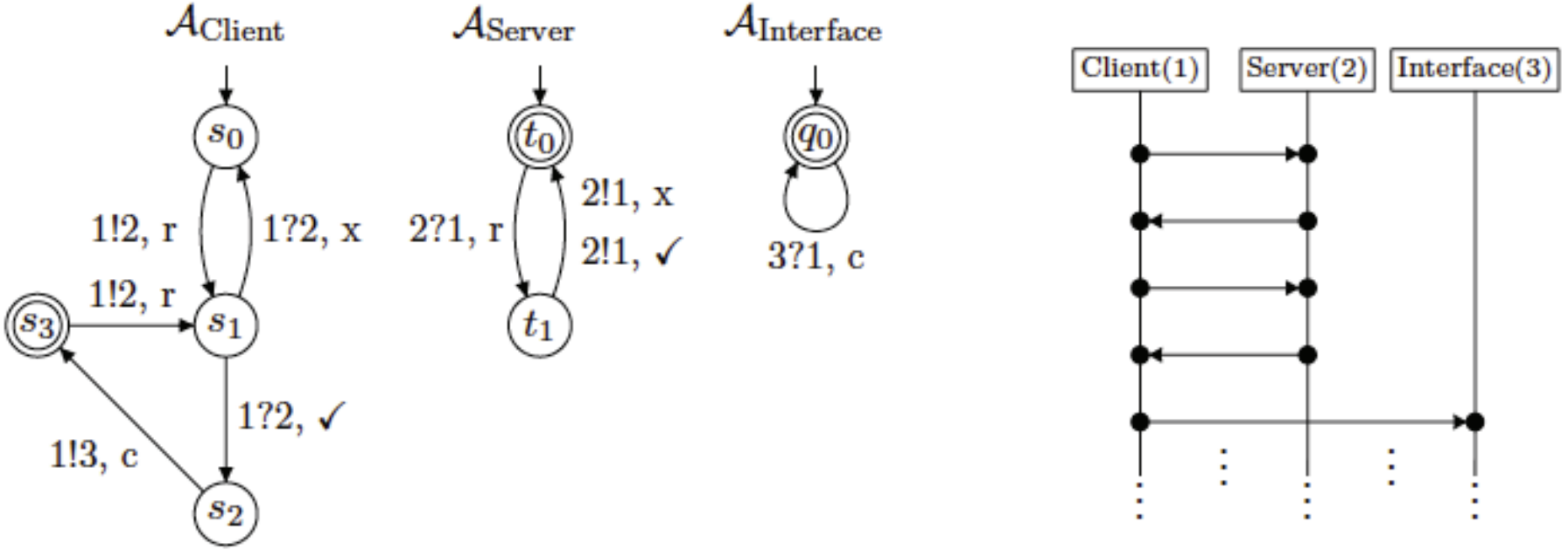
Overview

RWTH AACHEN UNIVERSITY

Communication finite-state machines

A CFM is accepting if all its processes have reached a local accepting state and either halt there or visit a local accepting state infinitely often.

An example CFM and an infinite MSC accepted by it



Client-server interaction to get access to an interface. Accepting state is (s_3, t_0, q_0).

PDL formulas on CFMs

A CFM is accepting if all its processes have reached a local accepting state and reside their ad infinitum.

The language $L(\mathcal{A})$ of CFM $\mathcal{A}$ is the set of MSCs that admit an accepting run.

CFM versus PDL

A CFM $\mathcal{A}$ satisfies PDL-formula Φ, denoted $\mathcal{A} \models \Phi$, whenever for all MSCs M it holds: $M \in L(\mathcal{A})$ if and only if $M \models \Phi$.

The example CFM satisfies $\forall\,(P_1 \rightarrow (\langle \mathsf{proc}^*; \mathsf{msg}; \mathsf{proc}^*; \mathsf{msg} \rangle\, P_3)$ where for $i \in \mathcal{P}$, formula $P_i = \bigvee_{j \in \mathcal{P}, j \neq i} (!_{i,j} \vee ?_{i,j})$, i.e., $M, e \models P_i$ iff e occurs at process i. The PDL formula asserts that process 3 (Interface) can be "reached" from 1 (Client) by exactly two messages using an intermediate process in between.

PDL model checking problem

Model checking CFMs versus PDL

The following model-checking problem is undecidable:

INPUT: a CFM $\mathcal{A}$, PDL-formula Φ

OUTPUT: is there an MSC $M \in L(\mathcal{A})$ with $M \models \Phi$?

Proof.

Follows immediately from the fact that the emptiness problem for CFMs is undecidable. By using the formula *true*, the above problem encodes the emptiness problem. □

To obtain decidability of the model-checking problem, we restrict ourselves to B-bounded MSCs.

RWTH AACHEN UNIVERSITY

PDL model checking problem

Model checking CFMs versus PDL [Bollig et. al, 2011]

The following model-checking problem is PSPACE-complete:

INPUT: a CFM $\mathcal{A}$ and $B \in \mathbb{N}_{>0}$, PDL-formula Φ

OUTPUT: is there an $\exists B$-bounded MSC $M \in L(\mathcal{A})$ with $M \models \Phi$?

Proof.

(Sketch). Every PDL formula Φ can effectively be translated into a CFM $\mathcal{A}_\Phi$ such that $\mathcal{A}_\Phi \models \Phi$. The details are out of the scope of this lecture. This synthesis step is independent of the channel bound size B (if any). The size of $\mathcal{A}_\Phi$ is exponential in the length of Φ and the number of processes in $\mathcal{P}$. Then construct a CFM accepting $L(\mathcal{A}) \cap L(\mathcal{A}_\Phi)$. Decide whether the resulting CFM accepts some $\exists B$-bounded MSC. This can all be done in polynomial space. The PSPACE-hardness follows from the hardness of LTL model checking. □

Model checking an MSG versus PDL

Model checking MSGs versus PDL [Bollig et. al, 2011]

The following model-checking problem is PSPACE-complete:

INPUT: a MSG G and PDL-formula Φ

OUTPUT: is there an MSC $M \in L(G)$ with $M \models \Phi$?

Proof.

(Sketch.) For every vertex v, we can determine a linearization of the MSC $\lambda(v)$. Construct a finite automaton $\mathcal{A}_G$ that accepts a linearization for every $M \in L(G)$, and vice versa, each word accepted by $\mathcal{A}_G$ is a linearization of some $M \in L(G)$. The size of $\mathcal{A}_G$ is linear in the size of G. Construct a CFM $\mathcal{A}_\Phi$ for PDL-formula Φ with $M \in L(\mathcal{A}_\Phi)$ iff $M \models \Phi$. Construct a transition system by running $\mathcal{A}_G$ and $\mathcal{A}_\Phi$ simultaneously. This construction terminates as $\mathcal{A}_G$ only accepts linearizations that are B-bounded (as every linearization of MSG G is $\exists B$-bounded by definition). Deciding whether some simultaneous run is accepting can be done in polynomial space. The PSPACE-hardness follows from the hardness of LTL model checking. □