Theoretical Foundations of the UML

Lecture 15+16: A Logic for MSCs

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

moves.rwth-aachen.de/teaching/ss-20/fuml/

June 15, 2020

Outline

Overview

1. Introduction
2. Local Formulas and Path Expressions
   - Syntax
   - Formal Semantics
3. PDL Formulas
4. Verification problems for PDL
   - Model checking MSCs
   - Model checking CFMs
   - Model checking MSGs
   - Satisfiability

A logic for MSCs

- This lecture will be devoted to a logic that is interpreted over MSCs

M has an event

$!(1,2,a)$

φ

$\neg \varphi$

$\varphi_1 \wedge \varphi_2$

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

- Based on propositional dynamic logic (PDL) [Fischer & Ladner, 1979]
  - combines easy-to-grasp concepts such as regular expressions and Boolean operators

+ modal logic

$\neg \quad \vee \quad \wedge \quad \Rightarrow$

$\langle \; \rangle$

$[\;]$

$\alpha ; \alpha$

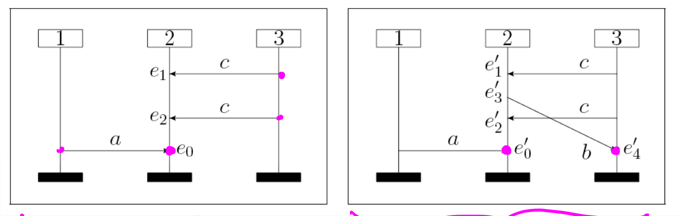
$\alpha + \alpha$

α^*

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
- Based on propositional dynamic logic (PDL) [Fischer & Ladner, 1979]
  - combines easy-to-grasp concepts such as regular expressions and Boolean operators
- Syntax, semantics, examples and various verification problems.

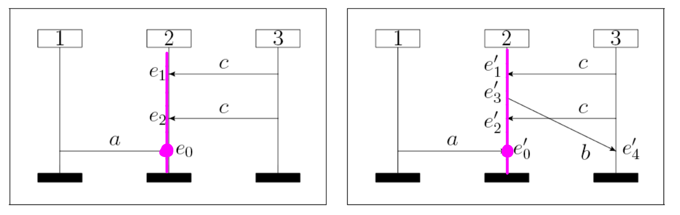
Some informal example properties



M_{left} M_{right} M_{left} M_{right}

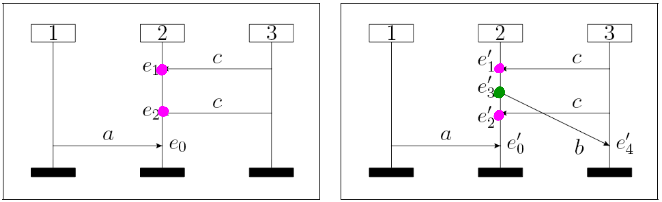
1. The (unique) maximal event of M is labeled by $?(2, 1, a)$ Yes. No.

Some informal example properties



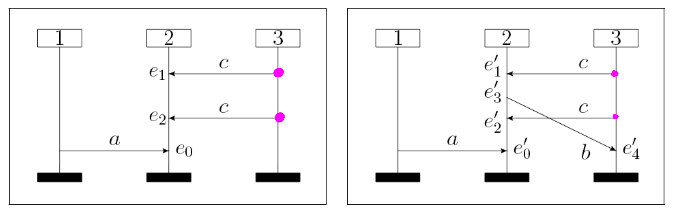
		M_{left}	M_{right}
1	The (unique) maximal event of M is labeled by $?(2,1,a)$	Yes.	No.
2	The maximal event on process 2 is labeled by $?(2,1,a)$	Yes.	Yes.

Some informal example properties



		M_{left}	M_{right}
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.

Some informal example properties



M_{left} M_{right}

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.

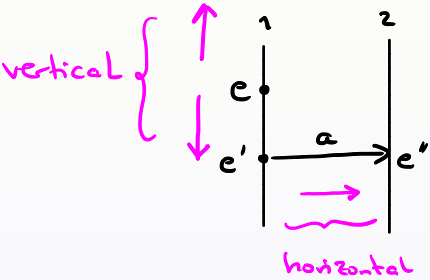
The need for logics

- Properties stated in natural language are ambiguous.
- We prefer to use a formal language for expressing properties.
- A formal semantics yields an unambiguous interpretation.
- This provides the basis for verification algorithms and common understanding.
- As formal language for properties we use logic.

Overview

The logic PDL

- Local formulas
  - Statements interpreted for single events in an MSC
  - Express properties about other events at the same process
  - Express properties about send and matched receive events



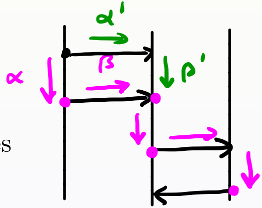
The logic PDL

- Local formulas
  - Statements interpreted for single events in an MSC
  - Express properties about other events at the same process
  - Express properties about send and matched receive events

- Path expressions
  - Used to navigate through an MSC
  - Use choice, concatenation and repetition
  - Can be embraced in box and diamond modalities

$+$ $\quad ;$ $\quad *$

regular expressions



$\langle \alpha ; \beta \rangle$ true

$\langle \alpha ; \beta + \alpha' ; \beta' \rangle$ true

The logic PDL

- Local formulas

  events

  - Statements interpreted for single events in an MSC
  - Express properties about other events at the same process
  - Express properties about send and matched receive events

- Path expressions
  - Used to navigate through an MSC
  - Use choice, concatenation and repetition
  - Can be embraced in box and diamond modalities

- PDL-formulas
  - Express properties about an entire MSC

Local formulas

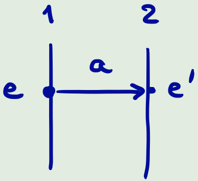
Local formulas

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 $!(1,2,a)$



$e \models\ !(1,2,a)$

$e' \not\models\ !(1,2,a)$

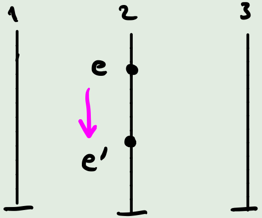
Local formulas

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 $!(1, 2, a)$
- $\langle \mathsf{proc} \rangle \mathit{true}$ — There is a next event at the same process



$e \models \langle proc \rangle \, true$

$e' \not\models \langle proc \rangle \, true$

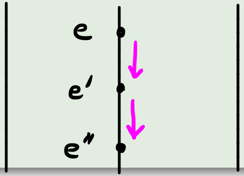
Local formulas

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 $!(1, 2, a)$
- $\langle \mathsf{proc} \rangle \mathit{true}$ — There is a next event at the same process
- $\langle \underline{\mathsf{proc}}; \underline{\mathsf{proc}} \rangle \mathit{true}$ — There are (at least) two next events at this process



$e \models \langle \mathsf{proc}; \mathsf{proc} \rangle \mathit{true}$

Local formulas

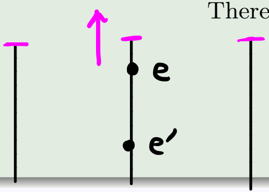
Local formulas

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 $!(1, 2, a)$
- $\langle \mathsf{proc} \rangle \mathit{true}$ — There is a next event at the same process
- $\langle \mathsf{proc}; \mathsf{proc} \rangle \mathit{true}$ — There are (at least) two next events at this process
- $[\mathsf{proc}]^{-1} \mathit{false}$ — There is no preceding event at this process

"look backwards"



$e \models [\mathsf{proc}]^{-1} \mathit{false}$

$e' \not\models [\mathsf{proc}]^{-1} \mathit{false}$

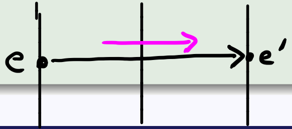
Local formulas

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 $!(1, 2, a)$
- $\langle \mathsf{proc} \rangle \mathit{true}$ — There is a next event at the same process
- $\langle \mathsf{proc}; \mathsf{proc} \rangle \mathit{true}$ — There are (at least) two next events at this process
- $[\mathsf{proc}]^{-1} \mathit{false}$ — There is no preceding event at this process
- $\langle \mathsf{msg} \rangle \mathit{true}$ — This event is a send matching a (next) receive event



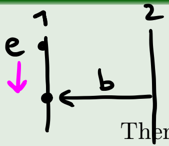
$e \models \langle \mathsf{msg} \rangle \mathit{true}$

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 $!(1, 2, a)$
$\langle \mathsf{proc} \rangle \, \mathit{true}$	There is a next event at the same process
$\langle \mathsf{proc}; \mathsf{proc} \rangle \, \mathit{true}$	There are (at least) two next events at this process
$[\mathsf{proc}]^{-1} \mathit{false}$	There is no preceding event at this process
$\langle \mathsf{msg} \rangle \, \mathit{true}$	This event is a send matching a (next) receive event
$\underline{\langle \mathsf{proc} \rangle} \; \underline{?(1, 2, b)}$	Event $?(1, 2, b)$ is a possible next event on this process

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 $!(1, 2, a)$
- $\langle \textsf{proc} \rangle \, true$ — There is a next event at the same process
- $\langle \textsf{proc}; \textsf{proc} \rangle \, true$ — There are (at least) two next events at this process
- $[\textsf{proc}]^{-1} false$ — There is no preceding event at this process
- $\langle \textsf{msg} \rangle \, true$ — This event is a send matching a (next) receive event
- $\langle \textsf{proc} \rangle \, ?(1, 2, b)$ — Event $?(1, 2, b)$ is a possible next event on this process
- $[\{ \neg !(1, 2, a) \}] \, true$ — An event is possible after any event different from $!(1, 2, a)$

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:

$$\varphi \ ::= \ \underline{true} \ \mid \ \underline{\sigma} \ \mid \ \underline{\neg\varphi} \ \mid \ \underline{\varphi \vee \varphi} \ \mid \ \underbrace{\langle\alpha\rangle\varphi}_{\text{"navigate" forwards}} \ \mid \ \underbrace{\langle\alpha\rangle^{-1}\varphi}_{\text{"navigate" backwards}}$$

The syntax of path expressions α will be defined later on.

$$\left.\begin{array}{l} !(1,2,a) \\ \neg\, ?(2,3,b) \end{array}\right\}$$

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:

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

The syntax of path expressions α will be defined later on.

Definition (Derived operators)

$$\left\{\begin{array}{rcl} false & := & \neg true \\ \varphi_1 \wedge \varphi_2 & := & \neg(\neg\varphi_1 \vee \neg\varphi_2) \\ \varphi_1 \rightarrow \varphi_2 & := & \neg\varphi_1 \vee \varphi_2 \\ [\alpha]\varphi & := & \neg\langle\alpha\rangle\neg\varphi \\ [\alpha]^{-1}\varphi & := & \neg\langle\alpha\rangle^{-1}\neg\varphi \end{array}\right.$$

De Morgan

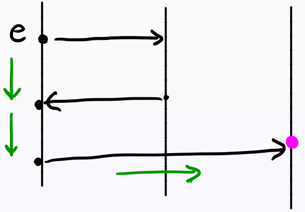
all possible α successors satisfy φ

there is no possible α-successor satisfying $\neg\varphi$

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

$\langle\,\rangle$

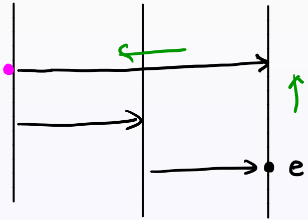


α = proc ; proc ; msg

φ = true

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



$\alpha =$ proc; msg

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

$\langle\alpha\rangle$ = "existential quantification over possible navigations through a MSC"

$[\alpha]$ = "universal quantification ......"

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?

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:

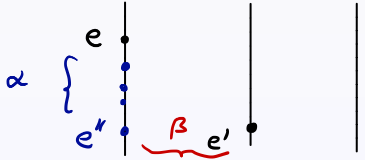
$$\alpha \ ::= \ \{\varphi\} \ \mid \ \mathsf{proc} \ \mid \ \mathsf{msg} \ \mid \ \underbrace{\alpha;\alpha}_{\text{concatenation}} \ \mid \ \underbrace{\alpha+\alpha}_{\text{alternative}} \ \mid \ \alpha^*$$

keywords

kleene star / repetition

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 $(e'', e') \models \beta$



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

zero or more times α

- α^* denotes the reflexive and transitive closure of the relation α

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 (p, say) 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., $\boxed{e' = m(e)}$

Forward and backward local formulas

- Event e satisfies $\langle \alpha \rangle \varphi$ iff there is an event e' such that (e, e') satisfies α and e' satisfies φ

  ↳ e' can be reached from e when navigating according to α.

Forward and backward local formulas

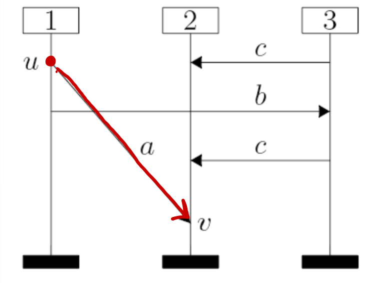
- Event e satisfies $\langle \alpha \rangle \varphi$ iff there is an event e' such that (e, e') satisfies α and e' satisfies φ

  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 (e', e) satisfies α and e' satisfies φ

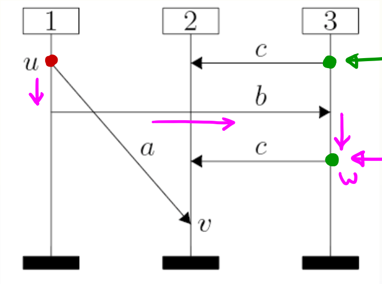
  Formula $\langle \alpha \rangle^{-1} \varphi$ looks "backward" along the partial order of the MSC starting from the current event

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 u's process
3. $u \models \langle \mathsf{msg} \rangle ?(2,1,a)$ — event u matches with the event v

$v \models ?(2,1,a)$



$\models \,!(3,2,c)$

$u \models \langle \text{proc}\,;\, \text{msg}\,;\, \text{proc} \rangle \overbrace{!(3,2,c)}^{\text{event } w}$

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 u's process
3. $u \models \langle \mathsf{msg} \rangle ?(2, 1, a)$ — event u matches with the event v
4. $u \models \langle \underbrace{(\mathsf{proc} + \mathsf{msg})^*}_{\alpha} \rangle !(3, 2, c)$ — event u happens before $!(3, 2, c)$

Semantics of local formulas (1)

Definition (Syntax of local formulas)

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

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

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 $((M, e), \varphi) \in \models$.

- $M, e \models true$ for all $e \in E$
- $M, e \models \sigma$ iff $l(e) = \sigma$
- $M, e \models \neg\varphi$ iff not$(M, e \models \varphi)$
- $M, e \models \varphi_1 \vee \varphi_2$ iff $M, e \models \varphi_1$ or $M, e \models \varphi_2$

Semantics of local formulas (2)

$\langle\alpha\rangle\varphi$

Definition (Semantics of forward path formulas)

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

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

$e \models (\langle\alpha\rangle)^{n+1}\varphi$ iff $e \models \langle\alpha\rangle(\langle\alpha\rangle^n)\varphi$ $\qquad (\langle\alpha\rangle)^0\varphi$ iff $e \models \varphi$

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 $<_p$.

$\langle\alpha\rangle^{-1}\varphi$

Definition (Semantics of backward path formulas)

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

- $e \models \langle\{\psi\}\rangle^{-1}\varphi$ iff $e \models \psi$ and $e \models \varphi$
- $e \models \langle\mathsf{proc}\rangle^{-1}\varphi$ iff $\exists e' \in E.\, \underline{e' <_p e}$ and $e' \models \varphi$
- $e \models \langle\mathsf{msg}\rangle^{-1}\varphi$ iff $\exists e' \in E.\, e' = \underline{m^{-1}(e)}$ and $e' \models \varphi$
- $e \models \langle\alpha_1; \alpha_2\rangle^{-1}\varphi$ iff $e \models \langle\alpha_1\rangle^{-1}\langle\alpha_2\rangle^{-1}\varphi$

$e \models \langle\alpha_1 + \alpha_2\rangle^{-1}\varphi$ iff $e \models \langle\alpha_1\rangle^{-1}\varphi$ or $e \models \langle\alpha_2\rangle^{-1}\varphi$

$e \models \langle\alpha^*\rangle^{-1}\varphi$ iff $\underline{\exists n \in \mathbb{N}}.\, e \models (\underline{\langle\alpha\rangle^{-1}})^n \varphi$

well-defined, as function m is bijective

Examples

$$M, e \models \langle proc \rangle \underbrace{\langle msg \rangle\ true}_{\varphi}$$

iff (* semantics of $\langle proc \rangle\, \varphi$ *)

$$\left(\exists \underset{\in E}{e'}.\ e \lessdot_p e' \wedge M, e' \models \langle msg \rangle \underbrace{true}_{\varphi'}\right)$$

iff (* semantics of $\langle msg \rangle\, \varphi'$ *)

$$\left(\exists \underset{\in E}{e'}.\ e \lessdot_p e' \wedge \left(\exists e'' \in E.\ e'' = m(e') \text{ and } e'' \models true\right)\right)$$

iff

$$\left(\exists e' \in E.\ e \lessdot_p e' \wedge \left(\exists e'' \in E.\ e'' = m(e')\right)\right)$$

"event e has a direct successor at its process which is $\underbrace{\text{a send event}}_{e'}$ with a matching receive"

Example 2

$$u \models [proc]^{-1} \text{ false}$$

iff (* rewrite $[\alpha]^{-1}$ in terms of $\langle\alpha\rangle^{-1}$ *)

$$u \models \neg \langle proc\rangle^{-1} \neg \text{ false}$$

iff (* semantics of $\neg$ *)

$$\text{not } (u \models \langle proc\rangle^{-1} \text{ true})$$

iff (* semantics of $\langle proc\rangle^{-1} \varphi$ *)

$$\text{not } (\exists v' \in E .\ v' \lessdot_p u \text{ and } v' \models \text{true})$$

iff

$$\text{not } (\exists v' \in E .\ v' \lessdot_p u)$$

"there is no preceding event to u at its process"

Example 3

$$u \models \langle \overbrace{(proc + msg)^*}^{} \rangle \overbrace{!(3,2,c)}^{\varphi}$$

iff (* semantics of $(\alpha)^*$ *)

$$\exists n \in \mathbb{N}.\ u \models \langle (proc + msg) \rangle^n\ !(3,2,c)$$

iff

$(\dagger)$ $\exists n \in \mathbb{N}.\ u \models \langle \underbrace{(proc + msg) ; \ldots ; (proc + msg)}_{n \text{ times}} \rangle !(3,2,c)$

In our example MSC of slide 16, this holds as for $n=3$:

$$u \models \langle proc ; msg ; proc \rangle\ !(3,2,c)$$

step-by-step this means

$$u \models \langle proc \rangle \underbrace{!(1,3,b)}_{e}$$

and $e \models \langle msg \rangle \underbrace{?(3,1,b)}_{e'}$

and $e' \models \langle proc \rangle \underbrace{!(3,2,c)}_{w}$

Overview

1. Introduction
2. Local Formulas and Path Expressions
   - Syntax
   - Formal Semantics
3. PDL Formulas
4. Verification problems for PDL
   - Model checking MSCs
   - Model checking CFMs
   - Model checking MSGs
   - Satisfiability

PDL formulas

properties over an entire MSC

Definition (Syntax of PDL formulas)

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

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

Negation

Negation is absent. As existential and universal quantification, as well as conjunction and disjunction are present, PDF-formulas are closed under negation.

Intuitive meaning of PDL formulas

- MSC M satisfies $\exists\varphi$ if M has some event e satisfying φ

Intuitive meaning of PDL formulas

- MSC M satisfies $\exists\varphi$ if M 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

Intuitive meaning of PDL formulas

- MSC M satisfies $\exists\varphi$ if M 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, every event that can be reached via an α-labelled path satisfies φ

Semantics of PDL formulas

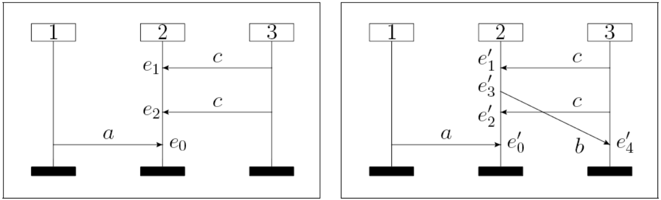
Definition (Semantics of PDL formulas)

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.

- $M \models \exists\varphi$ iff $\exists e \in E.\, M, e \models \varphi$
- $M \models \forall\varphi$ iff $\forall e \in E.\, M, e \models \varphi$
- $M \models \Phi_1 \wedge \Phi_2$ iff $M \models \Phi_1$ and $M \models \Phi_2$
- $M \models \Phi_1 \vee \Phi_2$ iff $M \models \Phi_1$ or $M \models \Phi_2$

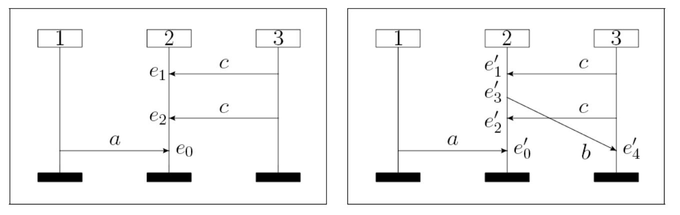
Example (1)



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

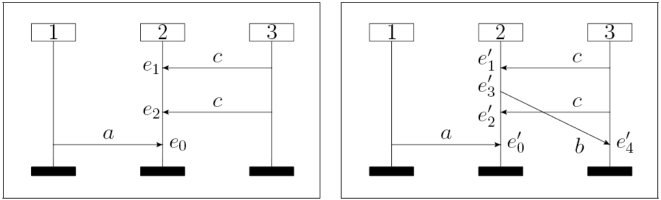
globally maximally

Example (2)



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

Example (3)



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

two consecutive events labeled ?(2,3,c)

impossible event