Overview

Theoretical Foundations of the UML

Lecture 1a: Introduction

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

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

April 20, 2020

Target audience

You are studying:

- Master Computer Science, or
- Master Data Science, or
- Master Systems Software Engineering, or
- Bachelor Computer Science, or
- ......

Usage as:

- elective course Theoretical Computer Science
- not a Wahlpflicht course for bachelor students
- specialization MOVES (Modeling and Verification of Software)
- complementary to Model-based Software Development (Rumpe)

Target audience (contd.)

In general:

- interest in system software engineering
- interest in formal methods for software
- interest in semantics and verification
- application of mathematical reasoning

Prerequisites:

- mathematical logic
- formal language and automata theory
- algorithms and data structures
- computability and complexity theory

regular languages
finite state automata

complexity classes
(NP, PSPACE)
undecidability

People involved

People involved:

	Lecturer	EMail
Lectures	Joost-Pieter Katoen	`katoen@cs.rwth-aachen.de`
Exercises	Mingshuai Chen	`chenms@cs.rwth-aachen.de`
	Bahare Salmani	`salmani@cs.rwth-aachen.de`

Organization

Schedule under the current COVID-19 circumstances:

- The lectures will take place in digital form (slide-casts)

  made available at lecture times

- The exercise classes will take place in digital form (slide-casts)
- A weekly Q&A session (on Thu, 16:00–17:30) via Zoom starting from April 23
- There will be about 21 lectures and 10 exercise classes
- Two lecture slide-casts per week starting from April 20 Mon Tue
- One exercise class slide-cast per week starting from April 27

Home assignments

Home assignments:

- weekly assignments: about 4 exercises to be solved by you
- groups of maximally three students together work on assignments
- solutions: hand in via RWTHmoodle[a] as pdf-file
- first assignment: Monday April 20 today !
- solution due at start next week: Monday April 27, 09:00
- first on-line exercise class video: Monday April 27 – explain solutions to the home assignments
- this scheme is repeated on a weekly basis until the beginning of July
- no lecture+exercise class in week following Pentecost excursion week

[a] You get access by enrolling to the exercise class via RWTHonline.

please do this asap !

Organization (contd.)

Examination: (6 ECTS credit points)

- written exam: July 23, 2020, 13:30-15:30 (Aula 2)
- written re-exam: September 2, 2020, 13:30–15:30 (Aula 2).

Details

- Admission: at least 40% of total amount of exercise points
- Registration: between May 1 and July 1 (via RWTHonline).

10 exercise classes of 100 points each

$\geq$ 400 points to be earned

Motivation

Scope:

- Goal: formal description + analysis of (concurr.) software systems
- Focus: the Unified Modeling Language

More specifically:

- Sequence Diagrams (used for requirements analysis)
- Propositional Dynamic Logic
- Communicating Finite State Automata
- Statecharts (behavioral description of systems)

Motivation

Scope:

- Goal: formal description + analysis of (concurr.) software systems
- Focus: the Unified Modeling Language

More specifically:

- Sequence Diagrams (used for requirements analysis)
- Propositional Dynamic Logic
- Communicating Finite State Automata
- Statecharts (behavioral description of systems)

Aims:

- clarify and make precise the semantics of some UML fragments
- formal reasoning about basic properties of UML models
- convince you that UML models are much harder than you think

What this course is **NOT** about:

What is it **not** about?

- the use of the UML in the software development cycle
  - see the complementary course by Prof. Rumpe
- other notations of the UML (e.g., class diagrams, activity diagrams)
- what is precisely in the UML, and what is not
  - liberal interpretation of which constructs belong to the UML
- applying the UML to concrete SW development case studies
- empirical results on the usage of UML
- drawing pictures
- ...

Overview

Theoretical foundations

sequence diagrams — statecharts

||

MSC

1. Lecture 1a: Introduction
2. Lecture 1b: Message Sequence Charts

Theoretical Foundations of the UML

Lecture 1b: Message Sequence Charts

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

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

April 20, 2020

History

- 1970s – 1980s: often used informally
- 1992: first version of MSCs standardized by CCITT (currently ITU) Z.120
- 1992 – 1996: many extensions, e.g., high-level + formal semantics (using process algebras)

  message sequence graphs
- 1996: MSC'96 standard

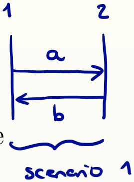
  included the semantics
- 2000: MSC 2000, time, data, o-o features
- 2005: MSC 2004
- 2011: latest standard published

Variants of MSCs

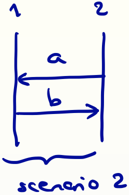
- UML sequence diagrams
- (instantiations of) use cases
- triggered MSCs
- netcharts (= Petri net + MSC)
- STAIRS
- Live sequence charts
- ...

extension of MSC
"hot" events
"cold" events

Characteristics



is a single scenario



- scenario-based language
- visual representation

  graphical
- "easy" to comprehend

  high-level MSCs
- generalization possible towards automata (states are MSCs)

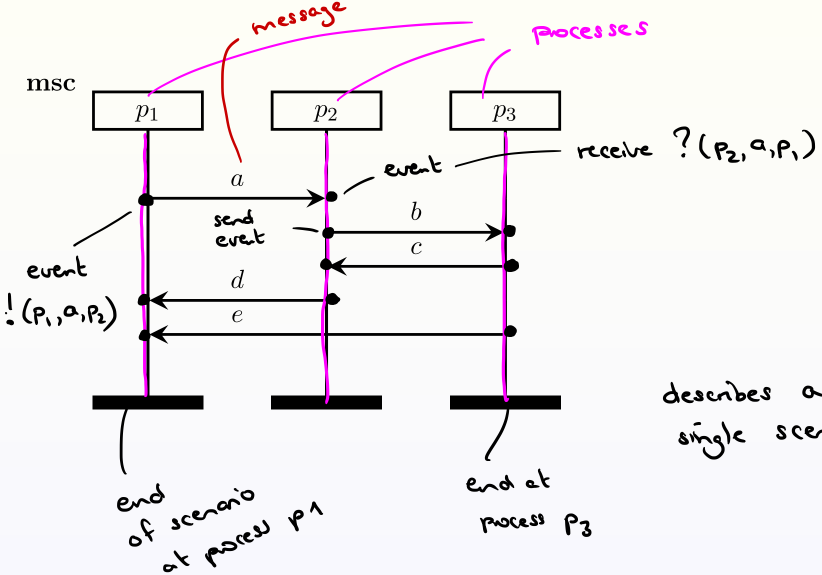
  message sequence graphs
- widely used in industrial practice

Applications

- requirements specification
  (positive, negative scenarios, e.g., CREWS)
- system design and software engineering
- visualization of test cases
  (graphical extension to TTCN)

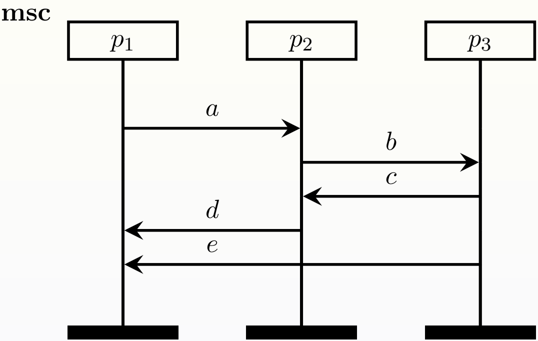
  standardised test notation
- feature interaction detection
- workflow management systems
- ...

Example



describes a single scenario

Example



These pictures are formalized using partial orders.

Partial orders

Definition

Let E be a set of events.
binary
A partial order over E is a relation $\preceq \subseteq \underbrace{E \times E}$ such that:

1. $\preceq$ is reflexive, i.e., $\forall e \in E.\, e \preceq e$,
2. $\preceq$ is transitive, i.e., $e \preceq e' \wedge e' \preceq e''$ implies $e \preceq e''$, and
3. $\preceq$ is anti-symmetric, i.e., $\forall e, e'.\, (e \preceq e' \;\wedge\; e' \preceq e) \Rightarrow e = e'$.

The pair $(E, \preceq)$ is called a partially ordered set (poset, for short).

<u>Example</u> E : sets of natural numbers e.g. $e \in E = \{0, 2, 7\}$
ordering $\subseteq$ $(E, \subseteq)$ is poset.

1. for every set e, $e \subseteq e$
2. $e \subseteq f$ and $f \subseteq g$ then $e \subseteq g$
3. $e \subseteq f$ and $f \subseteq e$ then $e = f$

E = bitstrings , e.g. $e \in E = 01011$

$e \leq f$ iff $\text{length}(e) \leq \text{length}(f)$

$(E, \leq)$ is not a poset.

1. $e \leq e$

2. $e \leq f$ and $f \leq g$ then $e \leq g$

3. $e \leq f$ and $f \leq e$, then

$\text{length}(e) \leq \text{length}(f)$ and

$\text{length}(f) \leq \text{length}(e)$

but not necessarily $e = f$.

Partial orders

Definition

Let E be a set of events.

A partial order over E is a relation $\preceq \, \subseteq E \times E$ such that:

1. $\preceq$ is reflexive, i.e., $\forall e \in E.\, e \preceq e$,
2. $\preceq$ is transitive, i.e., $e \preceq e' \wedge e' \preceq e''$ implies $e \preceq e''$, and
3. $\preceq$ is anti-symmetric, i.e., $\forall e, e'.\, (e \preceq e' \;\wedge\; e' \preceq e) \Rightarrow e = e'$.

The pair $(E, \preceq)$ is called a partially ordered set (poset, for short).

Definition

Let $(E, \preceq)$ be a poset and let $e, e' \in E$. e and e' are comparable if $e \preceq e'$ or $e' \preceq e$. Otherwise, they are incomparable.

set $\{1,2\}$ is incomparable to $\{3\}$, neither $\{3\} \subseteq \{1,2\}$
nor $\{1,2\} \subseteq \{3\}$

Partial orders

Definition

Let E be a set of events.

A partial order over E is a relation $\preceq \subseteq E \times E$ such that:

1. $\preceq$ is reflexive, i.e., $\forall e \in E.\, e \preceq e$,
2. $\preceq$ is transitive, i.e., $e \preceq e' \wedge e' \preceq e''$ implies $e \preceq e''$, and
3. $\preceq$ is anti-symmetric, i.e., $\forall e, e'.\, (e \preceq e' \;\wedge\; e' \preceq e) \Rightarrow e = e'$.

The pair $(E, \preceq)$ is called a partially ordered set (poset, for short).

Definition

Let $(E, \preceq)$ be a poset and let $e, e' \in E$. e and e' are comparable if $e \preceq e'$ or $e' \preceq e$. Otherwise, they are incomparable.

$\preceq$ is a non-strict partial order as it is reflexive. A strict partial order is a relation $\prec$ that is irreflexive, transitive and anti asymmetric (i.e., if $e \prec e'$ then not $e' \prec e$).

Hasse diagram

Definition

Let $(E, \preceq)$ be a poset.

The Hasse diagram $(E, \lessdot)$ of $(E, \preceq)$ is defined by:

$$e \lessdot e' \text{ iff } e \preceq e' \text{ and } \neg(\exists e'' \neq e, e'.\, e \preceq e'' \wedge e'' \preceq e')$$

Hasse diagrams can be used to visualize posets with finitely many elements in a succinct way.

Linearizations

Definition

Let $(E, \preceq)$ be a poset.
A linearization of $(E, \preceq)$ is a $\boxed{\text{total order}}$ $\sqsubseteq \subseteq E \times E$ such that

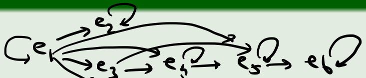
$$e \preceq e' \quad \text{implies} \quad e \sqsubseteq e'$$

A linearization is a topological sort of the Hasse diagram of $(E, \preceq)$.

Note that every partial order has at least one linearization.

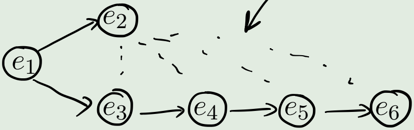
Example

Let $E = \{e_1, \ldots, e_6\}$,



$$
\begin{aligned}
\preceq \quad = \quad \{ \quad & (e_1, e_2), (e_1, e_3), (e_3, e_4), (e_4, e_5), (e_5, e_6), (e_1, e_4), \\
& (e_3, e_5), (e_1, e_5), (e_1, e_6), (e_3, e_6), (e_4, e_6) \\
\}^r \quad & \text{where } R^r \text{ denotes the reflexive closure of } R
\end{aligned}
$$

Hasse diagram:



Linearizations:

- $e_1 e_2 e_3 e_4 e_5 e_6$,
- $e_1 e_3 e_2 e_4 e_5 e_6$,
- $e_1 e_3 e_4 e_2 e_5 e_6$,
- $e_1 e_3 e_4 e_5 e_2 e_6$,
- $e_1 e_3 e_4 e_5 e_6 e_2$

$e \to e'$ means

$e \leq e'$

No linearizations:

- $e_2 e_1 e_3 \ldots$, and $e_1 e_4 e_3 \ldots$

Processes and actions

vertical lines

Definition

Let $\mathcal{P}$: finite set of (sequential) processes
$\mathcal{C}$: finite set of message contents $(a, b, c, \ldots \in \mathcal{C})$

Processes and actions

Definition

Let $\mathcal{P}$: finite set of (sequential) processes

$\mathcal{C}$: finite set of message contents $(a, b, c, \ldots \in \mathcal{C})$

Definition

Communication action: $p, q \in \mathcal{P}$, $p \neq q$, $a \in \mathcal{C}$

$!(p, q, a)$ "process p sends message a to process q"

$?(p, q, a)$ "process p receives message a sent by process q"

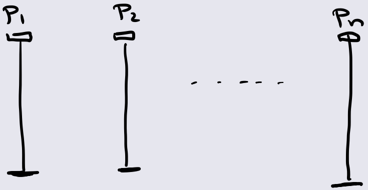
Let Act denote the set of communication actions (over $\mathcal{P}$ and $\mathcal{C}$)

Message Sequence Chart (MSC) (1)

Definition

An MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, \preceq)$ with:

- $\mathcal{P}$, a finite set of processes $\{p_1, p_2, \ldots, p_n\}$



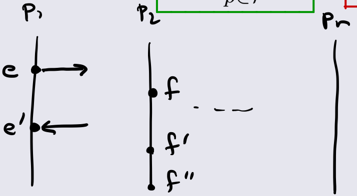
Definition

An MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, \preceq)$ with:

- $\mathcal{P}$, a finite set of processes $\{p_1, p_2, \ldots, p_n\}$ with $n > 1$
- E, a finite set of events

$$\boxed{E = \biguplus_{p \in \mathcal{P}} E_p} = \boxed{E_? \uplus E_!}$$

$E_{p_1} = \{e, e'\}$



$e \in E_!$

$e' \in E_?$

$E_{p_2} = \{f, f', f''\}$

Message Sequence Chart (MSC) (1)

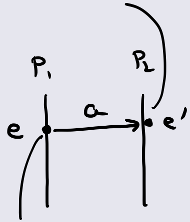
Definition

An MSC $M = (\underline{\mathcal{P}, E, \mathcal{C}, l}, m, \preceq)$ with:

$l(e') = ?(p_2, p_1, a)$

- $\mathcal{P}$, a finite set of processes $\{p_1, p_2, \ldots, p_n\}$ with $n > 1$
- E, a finite set of events

$$E = \biguplus_{p \in \mathcal{P}} E_p = E_? \uplus E_!$$



- $\mathcal{C}$, a finite set of message contents — a, b, c, ...
- $l : E \to Act$, a labelling function defined by:

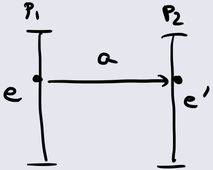
$l(e) = !(p_1, p_2, a)$

$$l(e) = \begin{cases} \underline{!(p, q, a)} & \text{if} \quad e \in \underline{E_p} \cap \underline{E_!} \\ \underline{?(p, q, a)} & \text{if} \quad e \in \underline{E_p} \cap \underline{E_?} \end{cases}, \text{for } p \neq q \in \mathcal{P},\ a \in \mathcal{C}$$

Definition

- $m : E_! \to E_?$ a bijection ("matching function"), satisfying:

$$m(e) = e' \wedge l(e) = !(p, q, a) \text{ implies } l(e') = ?(q, p, a) \quad (p \neq q,\ a \in \mathcal{C})$$



$m(e) = e'$

corresponding receive event of $!(p,q,a)$

$m^{-1}(e') = e$

Message Sequence Chart (MSC) (2)

Definition

- $m : E_! \to E_?$ a bijection ("matching function"), satisfying:

$$m(e) = e' \wedge l(e) = !(p, q, a) \text{ implies } l(e') = ?(q, p, a) \quad (p \neq q,\ a \in \mathcal{C})$$

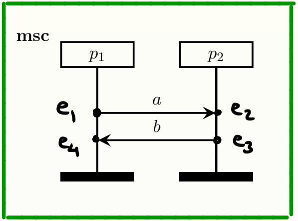
- $\preceq \subseteq E \times E$ is a partial order ("visual order") defined by:

$$\preceq \; = \Big(\underbrace{\bigcup_{p \in \mathcal{P}} <_p} \quad \cup \quad \underbrace{\{(\overset{!}{e}, m(\overset{?}{e})) \mid e \in E_!\}} \Big)^*$$

$<_p$ is a total order = "top-to-bottom" order on process p — communication order $<_c$

where for relation R, R^* denotes its reflexive and transitive closure.

Example (1)



$M = (\mathcal{P}, E, \mathcal{C}, l, m, \preceq)$ with:

- $\mathcal{P} \;=\; \{p_1, p_2\}$
- $E \;=\; \{e_1, e_2, e_3, e_4\}$
- $\mathcal{C} \;=\; \{a, b\}$
- $E_{p_1} = \{e_1, e_4\}$
- $E_{p_2} = \{e_2, e_3\}$
- $E_! = \{e_1, e_3\}$,
- $E_? = \{e_2, e_4\}$

- $l(e_1) = !(p_1, p_2, a)$
- $l(e_2) = ?(p_2, p_1, a)$
- $l(e_3) = !(p_2, p_1, b)$
- $l(e_4) = ?(p_1, p_2, b)$
- $m(e_1) = e_2$
- $m(e_3) = e_4$

Define $\preceq$:

- $e_1 < e_2$
- $e_3 < e_4$

Ordering at processes: $\underline{e_1 <_{p_1} e_4}$ and $\underline{e_2 <_{p_2} e_3}$.
Hasse diagram of $(E, \preceq)$:

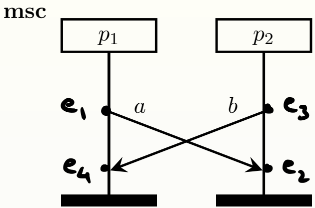
$$\boxed{e_1 \longrightarrow e_2 \longrightarrow e_3 \longrightarrow e_4}$$

total order

Linearizations?

$\{ e_1\, e_2\, e_3\, e_4 \}$

Example (2)



$M' = (\underbrace{\mathcal{P}, E, \mathcal{C}, l, m}_{\text{as above}}, \preceq')$ with:

$m(e_1) = e_2$

$m(e_3) = e_4$

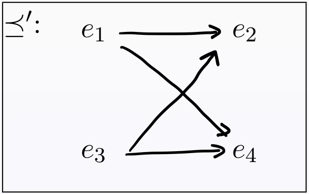
Hasse diagram of $\preceq'$:

① $<'_c$: $e_1 \longrightarrow e_2$

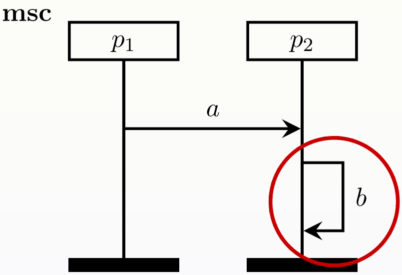
$e_3 \longrightarrow e_4$

② $<'_{p_1}$: $e_1 \longrightarrow e_4$

$<'_{p_2}$: $e_3 \longrightarrow e_2$



This is not an MSC



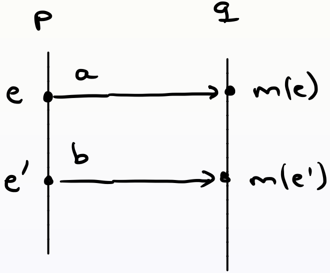
FIFO property

MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, \preceq)$ has the *First-In-First-Out* (FIFO) property whenever: for all $e, e' \in E_!$ we have

$$\boxed{e \prec e' \wedge l(e) = !(p,q,a) \wedge l(e') = !(p,q,b) \text{ implies } m(e) \prec m(e')}$$

i.e., "no message overtaking allowed"

$e \leq e' \wedge e \neq e'$

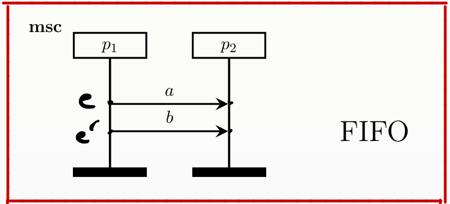


FIFO property

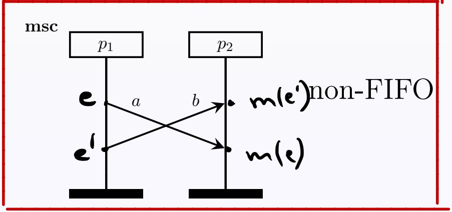
MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, \preceq)$ has the *First-In-First-Out* (FIFO) property whenever: for all $e, e' \in E_!$ we have

$$e \prec e' \wedge l(e) = !(p, q, a) \wedge l(e') = !(p, q, b) \text{ implies } m(e) \prec m(e')$$

i.e., "no message overtaking allowed"



$$\begin{aligned} l(e) &= !(p_1, p_2, a) \\ l(e') &= !(p_1, p_2, b) \\ e &\prec e' \\ \Rightarrow \quad m(e) &\prec m(e') \end{aligned}$$



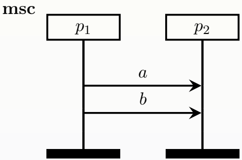
e < e' but

m(e') < m(e)

MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, \preceq)$ has the *First-In-First-Out* (FIFO) property whenever: for all $e, e' \in E_!$ we have

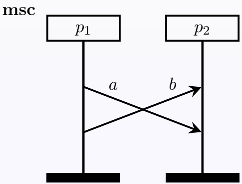
$$e \prec e' \wedge l(e) = !(p, q, a) \wedge l(e') = !(p, q, b) \text{ implies } m(e) \prec m(e')$$

i.e., "no message overtaking allowed"



FIFO

$$\begin{aligned} & l(e) = !(p_1, p_2, a) \\ & l(e') = !(p_1, p_2, b) \\ & e \prec e' \\ \Rightarrow \quad & m(e) \prec m(e') \end{aligned}$$



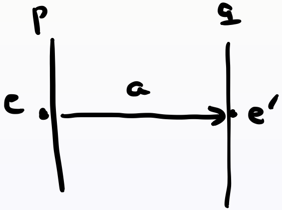
non-FIFO

Note:

We assume an MSC to possess the FIFO property, unless stated otherwise!

Definition

Let $Lin(M)$ = denote the set of (action) linearizations of MSC M.



$Lin(M) =$

$\{\ !(p,q,a)\ ?(q,p,a)\ \}$

$ee' \longmapsto \ell(e)\,\ell(e')$

$= \ !(p,q,a)\ ?(q,p,a)$

Linearizations

Definition

Let $Lin(M)$ = denote the set of (action) linearizations of MSC M.

! ?

$Lin(M)$ denotes a set of words over actions (and not over events)

the word of linearization $e_1 \ldots e_n$ equals $\ell(e_1) \ldots \ell(e_n)$

Linearizations

MSCs and its linearizations are interchangeable

There is a one-to-one correspondence between an MSC and its set of linearizations.

$Lin(M)$

MSC $M \longmapsto Lin(M)$

Linearizations

MSCs and its linearizations are interchangeable

There is a one-to-one correspondence between an MSC and its set of linearizations.

We will establish: the set $Lin(M)$ uniquely characterizes the MSC M (up to the event identities).

1 From MSCs to its set of linearizations is straightforward. — $Lin(M)$

2 The reverse direction is discussed in the following. First: well-formedness.

$w = \,!(p,q,a)\;?(q,p,a)\ldots\ldots$

Well-formedness

Let $Ch := \{(p,q) \mid p \neq q\,,\ p,q \in \mathcal{P}\}$ be the set of channels over $\mathcal{P}$.

We call $w = \underbrace{a_1 \ldots a_n \in Act^*}_{!(p,q,a)\ ?(q,p,a)\ \ldots}$ proper if

Let $Ch := \{(p,q) \mid p \neq q\ ,\ p,q \in \mathcal{P}\}$ be the set of channels over $\mathcal{P}$.

We call $w = a_1 \ldots a_n \in Act^*$ proper if

1. every receive in w is preceded by a corresponding send, i.e.:
   $\forall (p,q) \in Ch$ and prefix u of w, we have:

$$\underbrace{\sum_{m \in \mathcal{C}} |u|_{!(p,q,m)}}_{\#\text{ sends from } p \text{ to } q} \quad \geqslant \quad \underbrace{\sum_{m \in \mathcal{C}} |u|_{?(q,p,m)}}_{\#\text{ receipts by } q \text{ from } p}$$

where $|u|_a$ denotes the number of occurrences of action a in u

forbids $w = ?(q,p,a)\ !(p,q,a)$

$w = \underbrace{a_1 \ldots a_j}_{u} \ldots a_k$

Well-formedness

Let $Ch := \{(p,q) \mid p \neq q\ ,\ p,q \in \mathcal{P}\}$ be the set of channels over $\mathcal{P}$.

We call $w = a_1 \ldots a_n \in Act^*$ proper if

1. every receive in w is preceded by a corresponding send, i.e.:
   $\forall (p,q) \in Ch$ and prefix u of w, we have:

$$\underbrace{\sum_{m \in \mathcal{C}} |u|_{!(p,q,m)}}_{\#\text{ sends from } p \text{ to } q} \quad \geqslant \quad \underbrace{\sum_{m \in \mathcal{C}} |u|_{?(q,p,m)}}_{\#\text{ receipts by } q \text{ from } p}$$

   where $|u|_a$ denotes the number of occurrences of action a in u

2. the FIFO policy is respected, i.e.:
   $\forall 1 \leqslant i < j \leqslant n$, $(p,q) \in Ch$, and $a_i = \,!(p,q,m_1)$, $a_j = \,?(q,p,m_2)$:

$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} \;=\; \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)} \quad \text{implies} \quad m_1 = m_2$$

forbids eg. $w = \,!(p,q,a)\,!(p,q,b)\,?(q,p,b)\,?(q,p,a)$

Well-formedness

Let $Ch := \{(p, q) \mid p \neq q \, , \, p, q \in \mathcal{P}\}$ be the set of channels over $\mathcal{P}$.

We call $w = a_1 \ldots a_n \in Act^*$ proper if

1. every receive in w is preceded by a corresponding send, i.e.:
   $\forall (p, q) \in Ch$ and prefix u of w, we have:

$$\underbrace{\sum_{m \in \mathcal{C}} |u|_{!(p,q,m)}}_{\#\text{ sends from } p \text{ to } q} \quad \geqslant \quad \underbrace{\sum_{m \in \mathcal{C}} |u|_{?(q,p,m)}}_{\#\text{ receipts by } q \text{ from } p}$$

   where $|u|_a$ denotes the number of occurrences of action a in u

2. the FIFO policy is respected, i.e.:
   $\forall 1 \leqslant i < j \leqslant n$, $(p, q) \in Ch$, and $a_i = !(p, q, m_1)$, $a_j = ?(q, p, m_2)$:

no messages sent by p that are not received by q.

$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} \ = \ \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)} \quad \text{implies} \quad m_1 = m_2$$

no pending sent message

A proper word w is well-formed if $\boxed{\sum_{m \in \mathcal{C}} |w|_{!(p,q,m)} = \sum_{m \in \mathcal{C}} |w|_{?(q,p,m)}}$

$$w = \underbrace{!(r,q,m)\ !(p,q,m_1)\ !(p,q,m_2)\ ?(q,p,m_1)}_{u}\ ?(q,p,m_2)\ ?(q,r,m)$$

Claim: w is well-formed.

= w is proper + no "pending" messages

(1) consider e.g. prefix $u = !(r,q,m) \ldots ?(q,p,m)$

$$\underbrace{\sum |u|_{!(p,q,m)}}_{2} \geq \underbrace{\sum |u|_{?(q,p,m)}}_{1}$$

also $$\underbrace{\sum |u|_{!(r,q,m)}}_{1} \geq \underbrace{\sum |u|_{?(q,r,m)}}_{0}$$

check for all prefixes u of w.

(2) FIFO property:

for (p,q): m_1 is send and received before m_2

(r,q): trivially, as there is only 1 message

(3) every sent message in w is also received in w.

Properties of well-formedness

Proposition

For every MSC M and every $w \in Lin(M)$, w is well-formed.

<u>Lemma</u> $\forall$ MSC M. $\forall w \in Lin(M)$,

w is well-formed.

<u>Proof</u>: let $w = a_1 \dots a_n \in Lin(M)$

1. by definition if $a_j = ?(q, p, m)$ then $\exists a_i$ with $i<j$ $a_i = !(p, q, m)$. As this holds for every j it follows that for every prefix u of w and every $(p, q) \in Ch$:

$$\sum_m |u|_{!(p,q,m)} \geq \sum |u|_{?(q,p,m)}$$

2. by definition M is FIFO, thus w respects the FIFO property

3. for every $a_i = !(p, q, m)$, w contains a corresponding receive event $a_j = ?(q, p, m)$, as w is a linearization of MSC M $\boxtimes$

From linearizations to MSCs

$Lin(M) \rightsquigarrow M$

From linearizations to MSCs

Associate to $w = a_1 \ldots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

$E \longrightarrow Act$

From linearizations to MSCs

Associate to $w = a_1 \ldots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \ldots, n\}$ are the positions in w labelled with $\ell(i) = a_i$

From linearizations to MSCs

Associate to $w = a_1 \dots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \dots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\preceq = \left(\bigcup_{p \in \mathcal{P}} \prec_p \ \cup \ \prec_{\text{msg}} \right)^*$ where
  - $i \prec_p j$ if and only if $i < j$, for every $i, j \in E_p$

ordering on $\mathbb{N}$

From linearizations to MSCs

Associate to $w = a_1 \dots a_n \in Act^*$ an *Act*-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \dots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\preceq = \left(\bigcup_{p \in \mathcal{P}} \prec_p \ \cup \ \prec_{\text{msg}} \right)^*$ where
  - $i \prec_p j$ if and only if $i < j$, for every $i, j \in E_p$
  - $i \prec_{\text{msg}} j$ if for some $(p, q) \in Ch$ and $m \in \mathcal{C}$ we have:

$$\ell(i) = !(p, q, m) \text{ and } \ell(j) = ?(q, p, m) \text{ and}$$

$$\boxed{\sum_{m \in \mathcal{C}} |a_1 \dots a_{i-1}|_{!(p,q,m)} = \sum_{m \in \mathcal{C}} |a_1 \dots a_{j-1}|_{?(q,p,m)}}$$

$i<j$

From linearizations to MSCs

Associate to $w = a_1 \ldots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \ldots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\preceq = \left(\bigcup_{p \in \mathcal{P}} \prec_p \;\cup\; \prec_{\text{msg}} \right)^*$ where
  - $i \prec_p j$ if and only if $i < j$, for every $i, j \in E_p$
  - $i \prec_{\text{msg}} j$ if for some $(p, q) \in Ch$ and $m \in \mathcal{C}$ we have:

$$\ell(i) = !(p, q, m) \text{ and } \ell(j) = ?(q, p, m) \text{ and}$$

$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} \;=\; \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)}$$

From linearizations to MSCs

Associate to $w = a_1 \dots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \dots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\preceq = \left(\bigcup_{p \in \mathcal{P}} \prec_p \;\cup\; \prec_{\text{msg}} \right)^*$ where
  - $i \prec_p j$ if and only if $i < j$, for every $i, j \in E_p$
  - $i \prec_{\text{msg}} j$ if for some $(p, q) \in Ch$ and $m \in \mathcal{C}$ we have:

$$\ell(i) = !(p, q, m) \text{ and } \ell(j) = ?(q, p, m) \text{ and}$$
$$\sum_{m \in \mathcal{C}} |a_1 \dots a_{i-1}|_{!(p,q,m)} \;=\; \sum_{m \in \mathcal{C}} |a_1 \dots a_{j-1}|_{?(q,p,m)}$$

Example

construct $M(w)$ for $w = !(r, q, m)!(p, q, m_1)!(p, q, m_2)?(q, p, m_1)?(q, p, m_2)?(q, r, m)$

Example

construct $M(w)$ for $w = !(r,q,m)!(p,q,m_1)!(p,q,m_2)?(q,p,m_1)?(q,p,m_2)?(q,r,m)$

$a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6$

$E = \{e_1, \ldots, e_6\}$

$<_r : \quad \emptyset \qquad\qquad e_1 <_{msg} e_6$

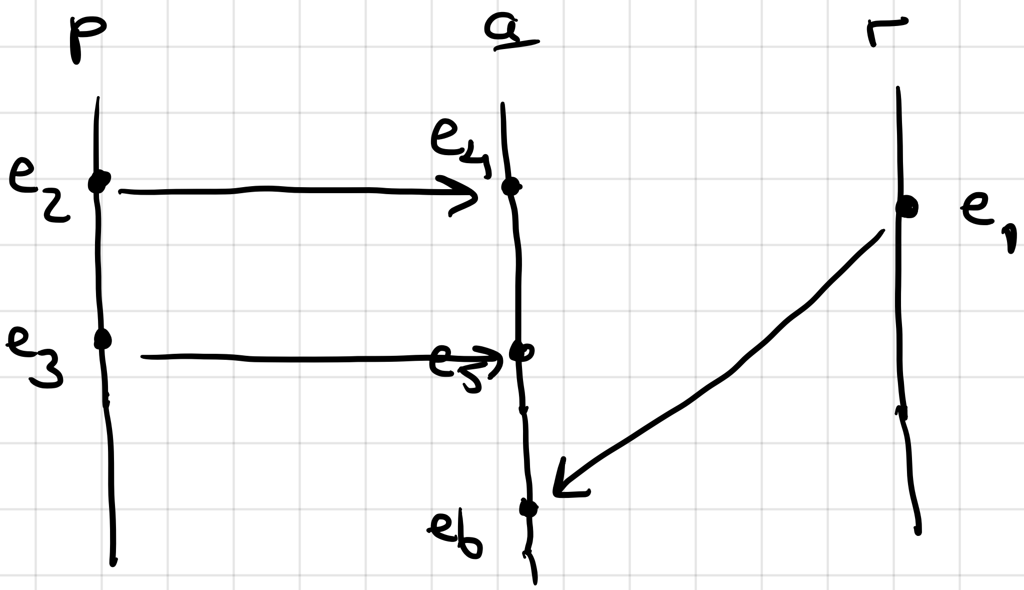
$<_p : \quad e_2 <_p e_3 \qquad\qquad e_2 <_{msg} e_4$

$<_q : \quad e_4 <_q e_5 \qquad\qquad e_3 <_{msg} e_5$

$e_5 <_q e_6$

$$\leq \; = \; \left(<_r \cup <_p \cup <_q \cup <_{msg} \right)^*$$

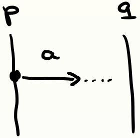
$M(w)$



From linearizations to MSCs

e.g. $w_0 = \,!(p,q,a)$ is not well-fomed

$M(w_0)$



not an MSC

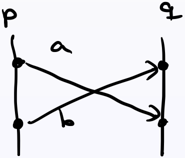
Relating well-formed words to MSCs

For every well-formed $w \in Act^*$, $M(w)$ is an MSC.

$w_1 = \,!(p,q,a)\;!(p,q,b)\;?(q,p,b)\;?(q,p,a)$

w_1 is not well-formed

$M(w_1)$



not FIFO, not MSC

Proof: for well-formed w, $M(w)$ is an MSC

(sketch). let w be $a_1 \dots a_n$ well-formed.

Construct $M(w)$ by a pass from left-to-right through w. let $w_k = a_1 \dots a_k$. Start with $w_0 = \varepsilon$, the empty word. Take $M(w)$ is empty labeled poset.

Now consider $w_{k+1} \neq \varepsilon$ and distinguish 2 cases:

① $w_{k+1} = w_k \, !(p, q, a)$. Then extend $M(w_k)$ with a new event e_{k+1} with $l(e_{k+1}) = !(p, q, a)$.

Extend means that all $e_i \in w_k \cap E_p$ precede e_{k+1} and that $m(e_{k+1})$ is undefined.

② $w_{k+1} = w_k \, ?(p, q, a)$. As w is well-formed, w_k is proper (by definition) thus $\exists a_i \in w_k$ with $a_j = !(q, p, a)$ for which $e_j \notin dom(m)$ in $M(w_k)$. Take the minimal j in $\{1, \dots, k\}$ with $e_j \notin dom(m)$. Extend $M(w_k)$ with e_{k+1}, $l(e_{k+1}) = ?(p, q, m)$ and $m(e_j) = e_{k+1}$

(3) As ω is well-formed it follows that for $k=n$, the function m is total on the set of send actions/events in w. $\boxtimes$

Definition

$(E, \preceq, \ell)$ and $(E', \preceq', \ell')$ are isomorphic if there exists a bijection $f : E \to E'$ such that $e \preceq e'$ iff $f(e) \preceq' f(e')$ and $\ell(e) = \ell'(f(e))$.

Linearizations yield isomorphic MSCs

For every well-formed $w \in Act^*$ and $w' \in Lin(M(w))$:

$$\boxed{M(w) \text{ and } M(w') \text{ are isomorphic.}}$$

w

$M(w) \rightsquigarrow w'$

$M(w')$

$M(w)$ and $M(w')$ are equal except for event identities.

Linearizations versus MSCs

Main theorem

There is a one-to-one relationship between MSC M and the set $Lin(M)$.