Foundations of the UML

Lecture 4: Properties of Message Sequence Graphs

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/i2/370

27. Oktober 2009

Message sequence graphs

Let $\mathbb{M}$ be the set of MSCs (up to isomorphism, i.e., event identities).

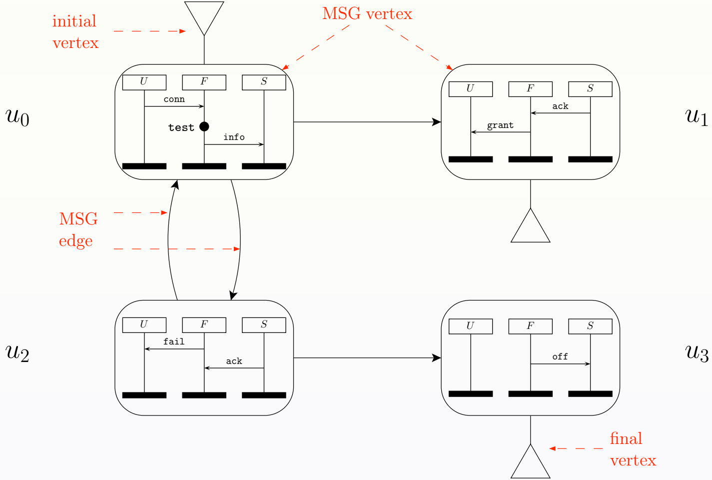
A Message Sequence Graph (MSG) G is a tuple $G = (V, \rightarrow, v_0, F, \lambda)$ with:

- $(V, \rightarrow)$ is a digraph with finite set V of vertices and $\rightarrow \subseteq V \times V$ a set of edges
- $v_0 \in V$ is the starting (or: initial) vertex
- $F \subseteq V$ is a set of final vertices
- $\lambda : V \rightarrow \mathbb{M}$ associates to each vertex $v \in V$, an MSC $\lambda(v)$

Note:

1. an MSG is an NFA without input alphabet where states are MSCs
2. every MSC is an MSG

Message sequence graphs



$$u_0\ u_2\ u_0\ u_1 = \lambda(u_0) \bullet \lambda(u_2) \bullet \lambda(u_0) \bullet \lambda(u_1)$$

Concatenation of MSCs

Let $M_i = (\mathcal{P}_i, E_i, \mathcal{C}_i, l_i, m_i, <_i) \qquad i \in \{1, 2\}$
be two MSCs with $E_1 \cap E_2 = \varnothing$

The concatenation of MSCs M_1 and M_2 is the MSC
$M_1 \bullet M_2 = (\mathcal{P}, E, \mathcal{C}, l, m, <)$ with:

$$\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \qquad E = E_1 \cup E_2 \qquad \mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$$

$$(\text{with } E_? = E_{1,?} \cup E_{2,?} \text{ etc.})$$

$$l(e) = \begin{cases} l_1(e) & \text{if} \quad e \in E_1 \\ l_2(e) & \text{if} \quad e \in E_2 \end{cases} \qquad m(e) = \begin{cases} m_1(e) & \text{if} \quad e \in E_1 \\ m_2(e) & \text{if} \quad e \in E_2 \end{cases}$$

$$< \,=\, <_1 \cup <_2 \cup \{(e, e') \mid \exists p \in \mathcal{P}.\, e \in E_1 \cap E_p\,,\; e' \in E_2 \cap E_p\}$$

MSC language of an MSG

Let $G = (V, \rightarrow, v_0, F, \lambda)$ be an MSG.

Definition

Path $\pi = u_0 \ \ldots \ u_n$ is accepting if: $u_0 = v_0$ and $u_n \in F$.

Definition

The MSC of a path $\pi = u_0 \ \ldots \ u_n$ is:

$$M(\pi) \ = \ \underbrace{\lambda(u_0)}_{\text{MSC of } u_0} \bullet \underbrace{\lambda(u_1)}_{\text{MSC of } u_1} \bullet \ldots \bullet \underbrace{\lambda(u_n)}_{\text{MSC of } u_n} \ = \ \prod_{i=0}^{n} \lambda(u_i)$$

Definition

The (MSC) language of MSG G is defined by:

$$L(G) = \{ M(\pi) \mid \pi \text{ is an accepting path of } G \}.$$

Facts about MSGs

Expressiveness

The state space of an MSG is context-sensitive.

Emptiness problem

Given MSGs G_1 and G_2, the problem to check whether $L(G_1) \cap L(G_2) = \varnothing$, is undecidable.

Local choice

Checking whether an MSG is local choice, is in PTIME.

Intersection of MSGs

Theorem

The decision problem:

$$\textit{for MSGs } G_1 \textit{ and } G_2, \textit{ do we have } L(G_1) \cap L(G_2) = \varnothing?$$

is undecidable.

Proof.

Reduction from Post's Correspondence Problem (PCP)

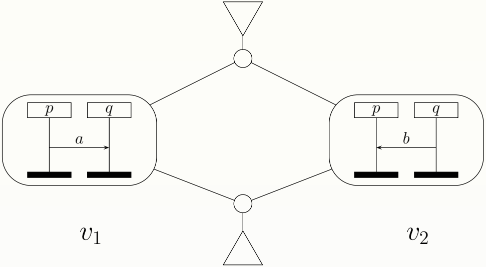
... black board ...

□

RWTH AACHEN UNIVERSITY

Local choice property (1)

G:



v_1 v_2

Inconsistency if process p behaves according to v_1
and process q behaves according to v_2

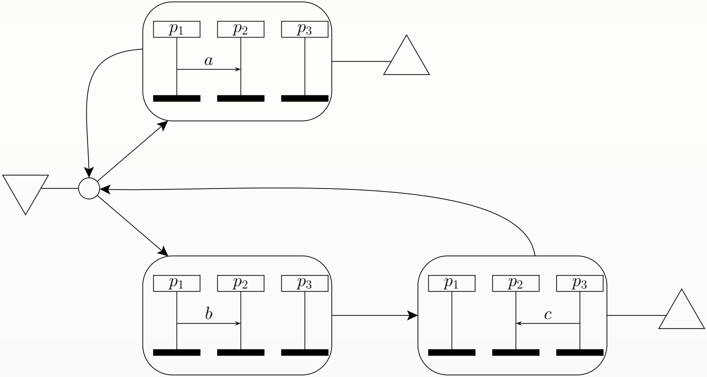
$\Longrightarrow$ possible distributed realization may yield a deadlock

Problem:

Subsequent behavior is determined by distinct processes

RWTHAACHEN UNIVERSITY

Example of local-choice MSG



Inconsistency if p_1 sends a and p_3 sends c.

Local choice property (2)

- e is a minimal event wrt. $\preceq$ if $\neg(\exists e' \neq e.\ e' \preceq e)$
- p is active in MSC M if $E_p \neq \varnothing$
- p is active in path $v_1 \ldots v_n$ in MSG G if p is active in $\lambda(v_i)$ for some i

Definition (local choice MSG)

MSG $G = (V, \rightarrow, v_0, F, \lambda)$ is local choice if:

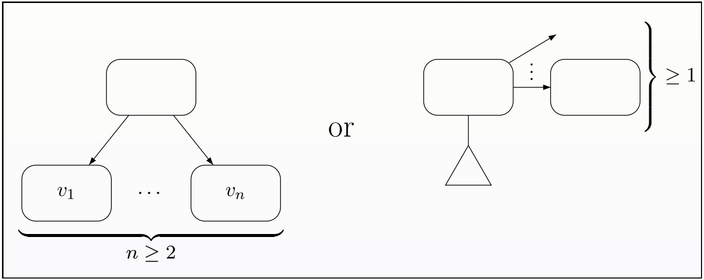
1. $\exists$ active p. $\forall \pi \in \text{Paths}(v_0)$.
   π contains a single minimal event $e \in E_p$
2. $\forall$ branching vertex $v \in V$. with $v \rightarrow w$
   $\exists$ active p . $\forall \pi \in \text{Paths}(w)$.
   π contains a single minimal event $e \in E_p$

Intuition:

Along every path from an initial or branching vertex there is a single process deciding how to proceed which can inform the other processes how to proceed.

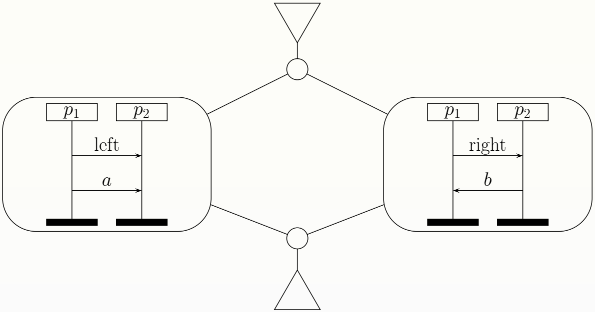
Branching vertices

A vertex is branching if:



Local choice

G:



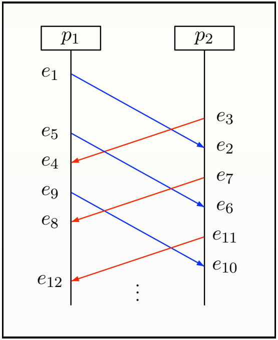
Note:

Checking whether an MSG is local choice can be done in PTIME.

How can non-local choice be resolved?

Refine your MSG and add control messages (cf. above example)

Restriction of MSGs [Yannakakis 1999]



This MSC cannot be decomposed as

$$M_1 \bullet M_2 \bullet \ldots \bullet M_n \quad \text{for } n > 1$$

This can be seen as follows:

- e_1 and $e_2 = m(e_1)$ must reside in same M_i
- $e_3 < e_2$ and $e_1 < e_4$ thus
  $e_3, e_4 \notin M_j$, $j < i$ or $j > i$
  $\implies e_3, e_4 \in M_i$
- by similar reasoning: $e_5, e_6 \in M_i$ etc.

Problem:

Compulsory matching between send and receive in same MSG vertex (i.e., send e and receive $m(e)$)

Compositional MSCs [Gunter, Muscholl, Peled 2001]

Solution: drop restriction that e and $m(e)$ belong to the same MSC (= allow for incomplete message transfer)

Definition

$M = (\mathcal{P}, E, \mathcal{C}, l, m, <)$ is a compositional MSC (CMSC, for short) where $\mathcal{P}, E, \mathcal{C}$ and l are as before, and

- $m : E_! \to E_?$ is a partial, injective function such that (as before):

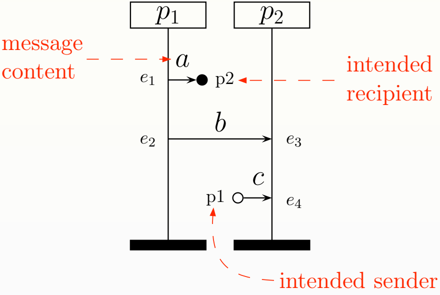
$$m(e) = e' \wedge l(e) = !(p, q, a) \implies l(e') = ?(q, p, a)$$

- $$< \; = \Big(\bigcup_{p \in \mathcal{P}} <_p \quad \cup \quad \{ (e, m(e)) \mid \underbrace{e \in \underbrace{dom(m)}_{\text{domain of } m}}_{\text{"}m(e)\text{ is defined"}} \} \Big)^*$$

Note:

An MSC is a CMSC where m is total and bijective.

CMSC example



$m(e_2) = e_3$
$e_1 \notin dom(m)$
$e_4 \notin rng(m)$

Definition

A compositional MSG (CMSG) $G = (V, \rightarrow, v_0, F, \lambda)$ with $\lambda : V \rightarrow \mathbb{CM}$, where $\mathbb{CM}$ is the set of all CMSCs, and $V, \rightarrow, v_0,$ and F as before.

Concatenation of CMSCs (1)

Let $M_i = (\mathcal{P}_i, E_i, \mathcal{C}_i, l_i, m_i, <_i) \in \mathbb{CM} \quad i \in \{1, 2\}$
be CMSCs with $E_1 \cap E_2 = \varnothing$

The concatenation of CMSCs M_1 and M_2 is the CMSC
$M_1 \bullet M_2 = (\mathcal{P}_1 \cup \mathcal{P}_2,\ E,\ \mathcal{C}_1 \cup \mathcal{C}_2, l, m, <)$ with:

- $E = E_1 \cup E_2$
- $l(e) = l_1(e)$ if $e \in E_1$, $l_2(e)$ otherwise
- $m(e) = E_! \rightarrow E_?$ satisfies:
  1. m extends m_1 and m_2, i.e., $e \in dom(m_i)$ implies $m(e) = m_i(e)$
  2. m matches unmatched send events in M_1 with unmatched receive events in M_2 according to order on process (matching from top to bottom)
     the k-th unmatched send in M_1 is matched with
     the k-th unmatched receive in M_2 (of the same "type")
  3. $M_1 \bullet M_2$ is FIFO (when restricted to matched events)

RWTHAACHEN UNIVERSITY

Concatenation of CMSCs (2)

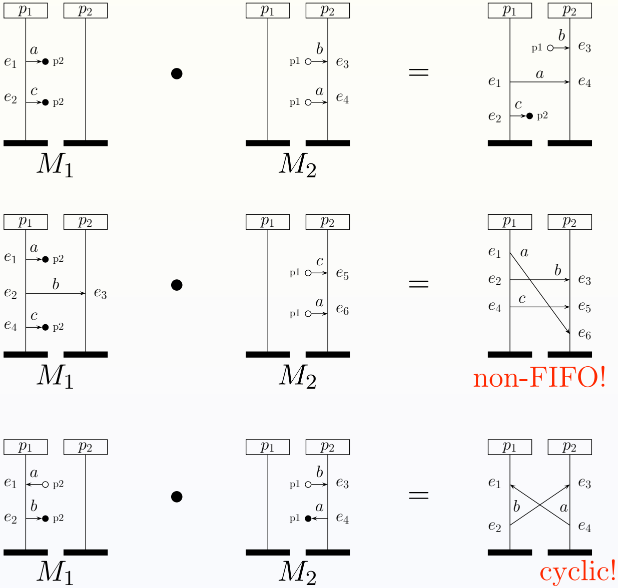
Let $M_i = (\mathcal{P}_i, E_i, \mathcal{C}_i, l_i, m_i, <_i) \in \mathbb{CM} \qquad i \in \{1,2\}$
be CMSCs with $E_1 \cap E_2 = \varnothing$

The concatenation of CMSCs M_1 and M_2 is the CMSC
$M_1 \bullet M_2 = (\mathcal{P}_1 \cup \mathcal{P}_2, E_1 \cup E_2, \mathcal{C}_1 \cup \mathcal{C}_2, l, m, <)$ with:

- $<$ is the reflexive and transitive closure of:

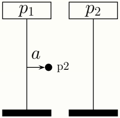
$$\left(\bigcup_{p \in \mathcal{P}} <_{p,1} \cup <_{p,2} \right) \cup \{(e, e') \mid e \in E_1 \cap E_p\,,\ e' \in E_2 \cap E_p\} \cup \{(e, m(e) \mid e \in dom(m)\}$$

Examples

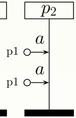


p1 p2
e1 a p2
e2 c p2
M1
•
p1 p2
b p1 e3
a p1 e4
M2
=
p1 p2
b p1 e3
e1 a e4
e2 c p2
p1 p2
e1 a p2
e2 b e3
e4 c p2
M1
•
p1 p2
c p1 e5
a p1 e6
M2
=
p1 p2
e1 a
e2 b e3
e4 c e5
e6
non-FIFO!
p1 p2
e1 a p2
e2 b p2
M1
•
p1 p2
b p1 e3
a p1 e4
M2
=
p1 p2
e1 e3
b a
e2 e4
cyclic!

Associativity

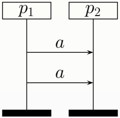


M

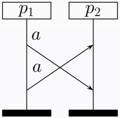


M'

$(M \bullet M) \bullet M'$:



$M \bullet (M \bullet M')$:



$\Longrightarrow$ this is non-FIFO (and thus undefined)

Note:

Concatenation of CMSCs is <u>not</u> associative.

Paths

Let $G = (V, \to, v_0, F, \lambda)$ be a CMSG.

Definition

A path π of G is a finite sequence

$$\pi = u_0\ u_1\ \ldots\ u_n \text{ with } u_i \in V\ \ (0 \leq i \leq n) \text{ and } u_i \to u_{i+1}\ \ (0 \leq i < n)$$

Definition

Path $\pi = u_0\ \ldots\ u_n$ is accepting if: $u_0 = v_0$ and $u_n \in F$.

Definition

The CMSC of a path $\pi = u_0\ \ldots\ u_n$ is:

$$M(\pi)\ =\ (\ldots(\lambda(u_0) \bullet \lambda(u_1)) \bullet \lambda(u_2) \ldots) \bullet \lambda(u_n)\ =\ \prod_{i=0}^{n} \lambda(u_i)$$

where CMSC concatenation is left assiociative.

Language of a CMSG

Definition

The (MSC) language of CMSG G is defined by:

$$L(G) = \{ \underbrace{M(\pi) \in \mathbb{M}}_{\text{only MSCs are considered}} \mid \pi \text{ is an accepting path of } G \}.$$

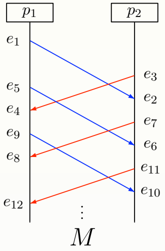
Definition (safeness)

CMSG G is safe if for every accepting path π of G, $M(\pi)$ is an MSC.

So:

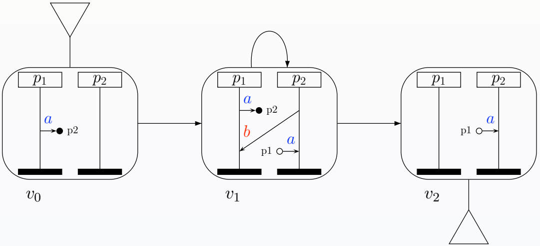
CMSG G is safe if on any of its accepting paths there are no unmatched sends and receipts.

Consider again



Recall: this behavior cannot be modeled for $n > 1$ by:

$$M = M_1 \bullet M_2 \bullet \ldots \bullet M_n \quad \text{with} \quad M_i \in \mathbb{M}$$



The (safe) CMSG for the above MSC.

RWTH AACHEN UNIVERSITY