Theoretical Foundations of the UML

Lecture 13: Local Choice MSGs and Regular Expressions on MSCs

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

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

June 8, 2020

Outline

Overview

Safe realisability

Definition (Realisability of MSGs)

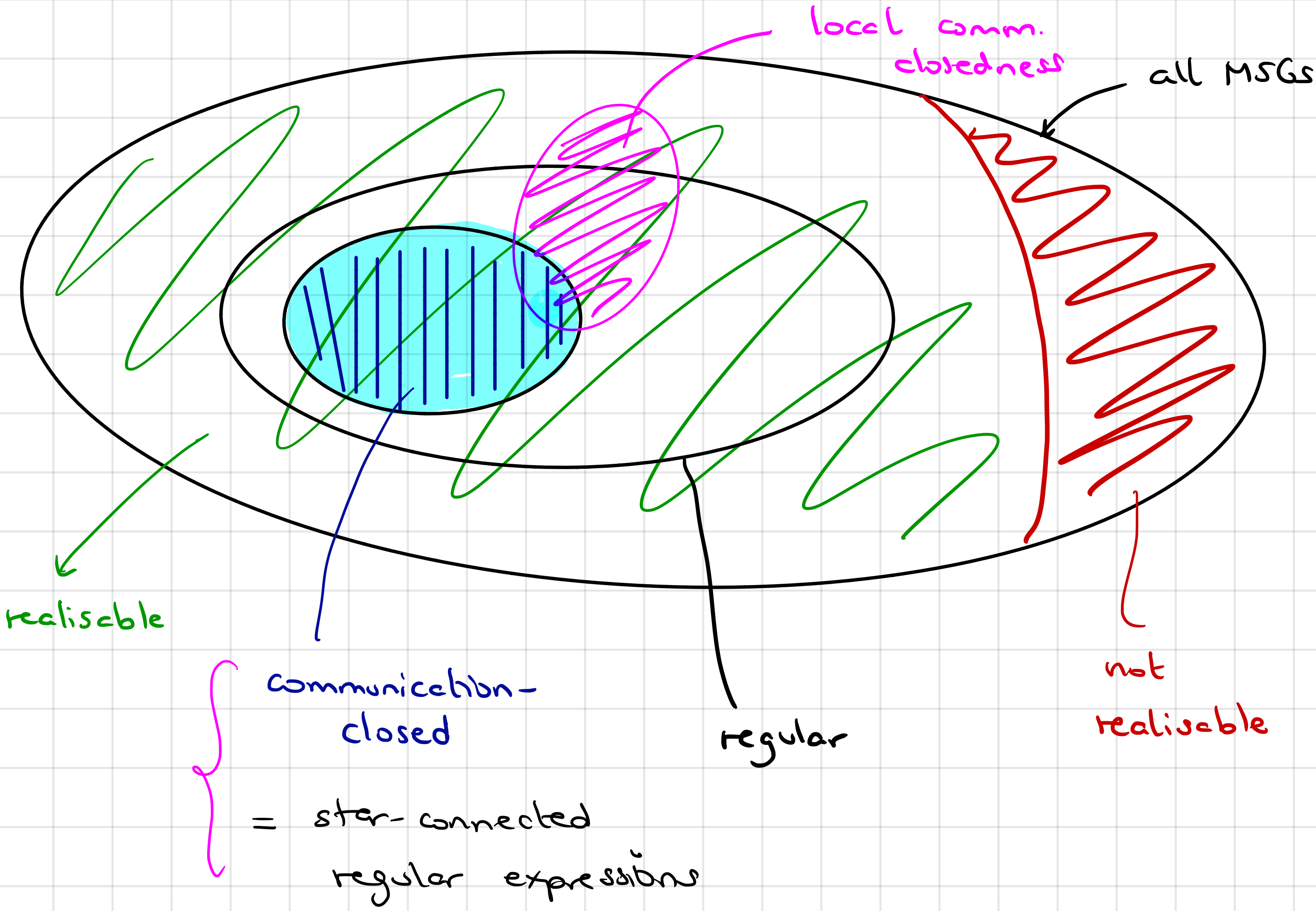
1. MSG G is realisable whenever $\mathcal{L}(G) = \mathcal{L}(\mathcal{A})$ for some CFM $\mathcal{A}$.
2. MSG G is safely realisable whenever $\mathcal{L}(G) = \mathcal{L}(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.

Summary of results

necessary + sufficient

Results so far:

1. Conditions for (safe) realisability for finite sets of MSCs.
2. Checking these conditions is co-NP complete (in P).
3. Regular MSGs are (safely) realisable by ∀-bounded CFMs.
4. Checking regularity of MSGs is undecidable.
5. Communication-closedness implies regularity; its check is co-NP complete.
6. Local communication-closedness implies realizability, and can be checked in P.


local comm. closedness
all MSGs
realisable
communication-closed
regular
not realisable
= star-connected regular expressions

Some remaining questions

- Can results be obtained for other classes of MSGs?
- What happens if we allow synchronisation messages?
  - recall that weak CFMs do not involve synchronisation messages
- How do we obtain a CFM realising an MSG algorithmically?
  - in particular, for local choice MSGs

Today's topics

The next two lectures

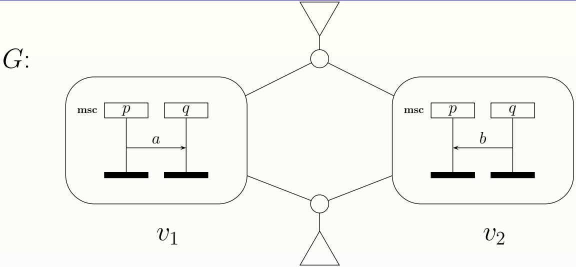
Safe realisability of (a somewhat restricted class of) MSGs. So as to obtain deadlock-free CFMs, the input MSG is required to be local choice. The CFMs are no longer weak. They exploit synchronisation messages.

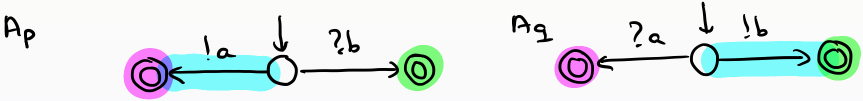
Results:

1. Realisability for certain regular expressions of local-choice MSGs.
2. An algorithm that generates a CFM from such local-choice MSG.

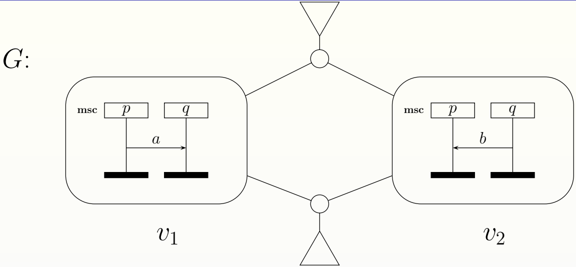
Overview

Non-local choice





Non-local choice



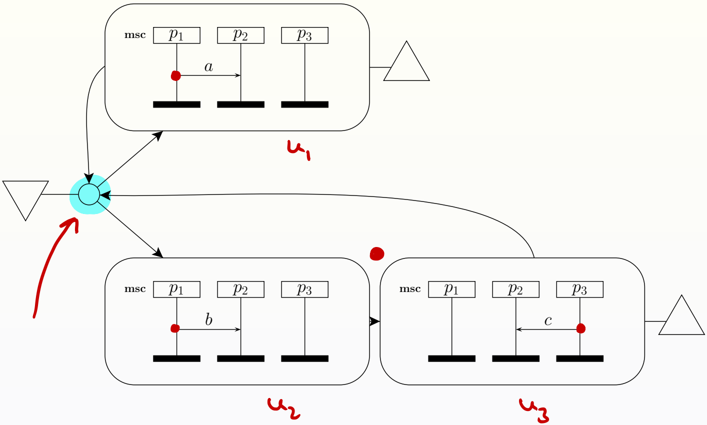
Inconsistency if process p behaves according to vertex v_1 and process q behaves according to vertex v_2

$\Longrightarrow$ realisation by a CFM may yield a deadlock

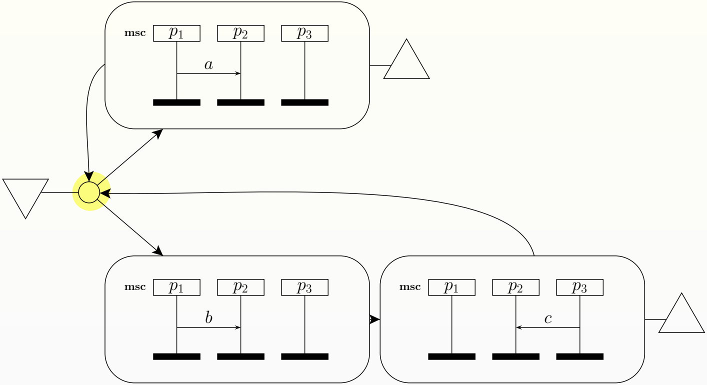
Problem:

Subsequent behavior in G is determined by distinct processes. When several processes independently decide to initiate behavior, they might start executing different successor MSCs (= vertices). This is called a non-local choice.

A (more involved) non-local choice

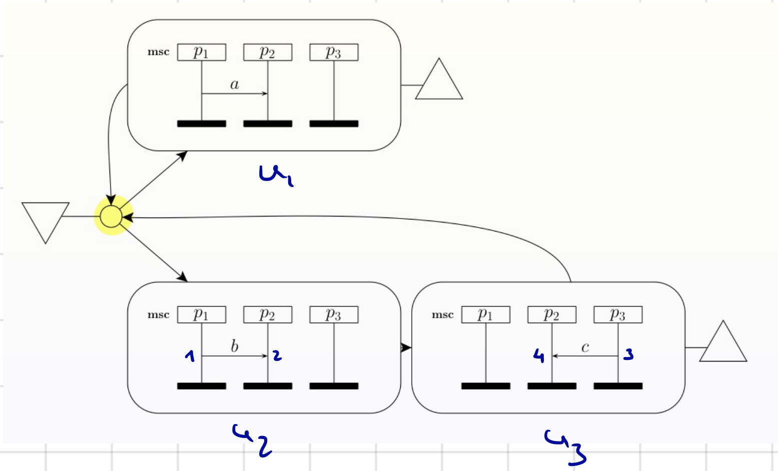


A (more involved) non-local choice



Problem:

Inconsistency if p_1 decides to send a and p_3 decides to send c.
Which branch to take in the initial vertex?



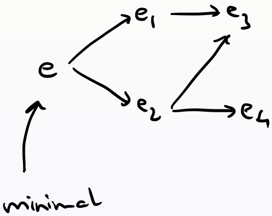
$\pi = u_2 u_3$ $\qquad$ $\min(\pi) = \{1, 3\}$

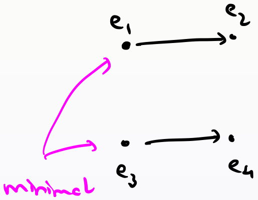
$1 \rightarrow 2$

$3 \rightarrow 4$

Definition (Minimal event)

Let $(E, \preceq)$ be a poset. Event $e \in E$ is a minimal event wrt. $\preceq$ if $\neg(\exists e' \neq e.\ e' \preceq e)$.





Preliminaries

Definition (Minimal event)

Let $(E, \preceq)$ be a poset. Event $e \in E$ is a minimal event wrt. $\preceq$ if $\neg(\exists e' \neq e.\ e' \preceq e)$.

Intuition: there is no event that has to happen before e happens. That is to say: the occurrence of e does not depend on any other event.

Definition (Partial order of a path)

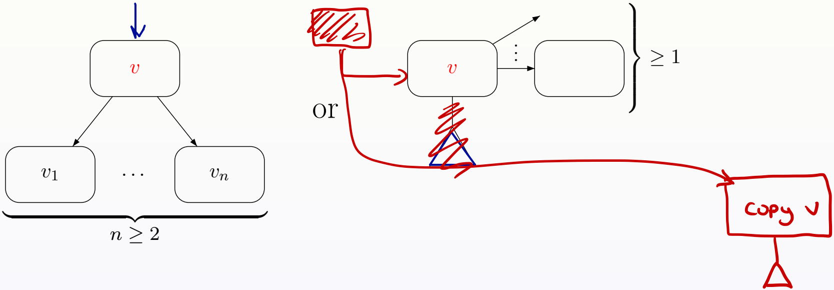
For finite path $\pi = v_1 \ldots v_n$ in MSG G, let $<_{M(\pi)}$ be the partial order of the MSC $M(\pi) = \lambda(v_1) \bullet \ldots \bullet \lambda(v_n)$.

Let $\min(\pi)$ be the set of minimal events wrt. $<_{M(\pi)}$ along finite path π.

Branching vertices

A branching vertex in MSG G either has at least two successors, or is a final vertex with at least one successor.

Pictorially, vertex v is branching if either:

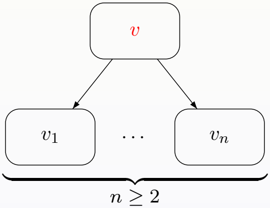


Without loss of generality we assume that branching final vertices do not occur.

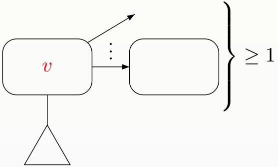
Branching vertices

A branching vertex in MSG G either has at least two successors, or is a final vertex with at least one successor.

Pictorially, vertex v is branching if either:



or



Without loss of generality we assume that branching final vertices do not occur. They can be always be removed at the expense of copying such vertices.

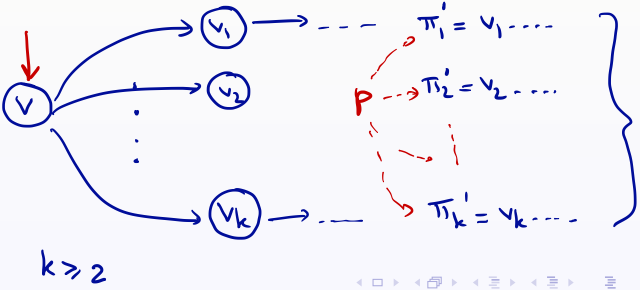
Local choice property

Definition (Local choice)

Let MSG $G = (V, \rightarrow, v_0, F, \lambda)$. MSG G is local choice if for every branching vertex $v \in V$ it holds:

$$\exists \text{process } p.\ \left(\forall \pi \in \text{Paths}(v).\ |\min(\pi')| = 1 \wedge \min(\pi') \subseteq E_p\right)$$

where for $\pi = v v_1 v_2 \ldots v_n$ we have $\pi' = v_1 v_2 \ldots v_n$.



Local choice property

Definition (Local choice)

Let MSG $G = (V, \rightarrow, v_0, F, \lambda)$. MSG G is local choice if for every branching vertex $v \in V$ it holds:

$$\exists \text{process } p. \ \big(\forall \pi \in \text{Paths}(v). \ |\min(\pi')| = 1 \ \wedge \ \min(\pi') \subseteq E_p\big)$$

where for $\pi = v v_1 v_2 \ldots v_n$ we have $\pi' = v_1 v_2 \ldots v_n$.

Intuition:

There is a single process that initiates behavior along every path from the branching vertex v.

Local choice property

Definition (Local choice)

Let MSG $G = (V, \rightarrow, v_0, F, \lambda)$. MSG G is local choice if for every branching vertex $v \in V$ it holds:

$$\exists \text{process } p. \ \big(\forall \pi \in \text{Paths}(v). \ |\min(\pi')| = 1 \ \wedge \ \min(\pi') \subseteq E_p\big)$$

where for $\pi = v v_1 v_2 \ldots v_n$ we have $\pi' = v_1 v_2 \ldots v_n$.

Intuition:

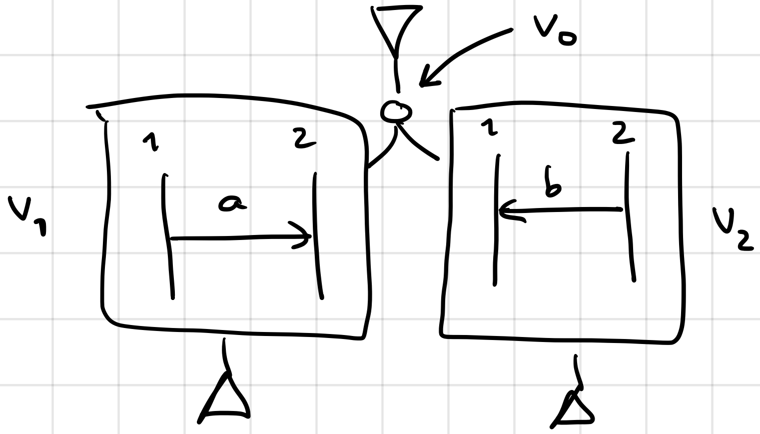
There is a single process that initiates behavior along every path from the branching vertex v. This process decides how to proceed. In a realisation by a CFM, it can inform the other processes how to proceed.

Local choice or not?

Deciding whether MSG G is local choice or not is in P. (Exercise.)

local choice MSG

G_1

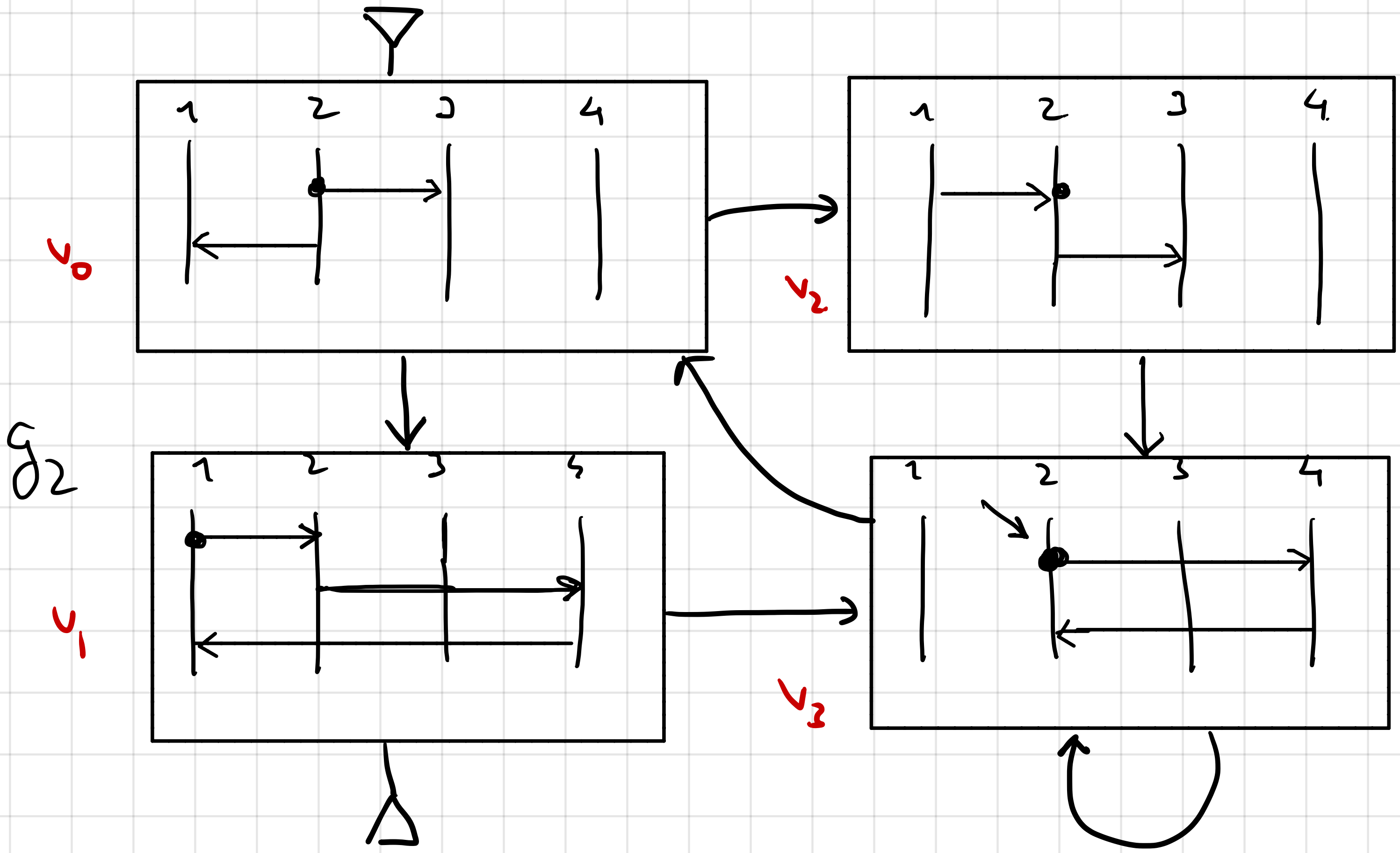


G_1 is non-local choice $Paths(v_0) = \{\underbrace{v_0 v_1}_{\pi_1}, \underbrace{v_0 v_2}_{\pi_2}\}$

$\pi_1' = v_1$ $min(\pi_1') = !a \longrightarrow$ process 1

$\neq \Rightarrow$ non local choice.

$\pi_2' = v_2$ $min(\pi_2') = !b \longrightarrow$ process 2



Claim: G_2 is local choice

branching vertices $= \{v_0, v_3\}$

a. v_0: $\pi_1 = v_0 \underline{v_1 v_3}$ $\min(\pi_1') = !(1,3)$

$\pi_2 = v_0 \underline{v_2 v_3} \dots$ $\min(\pi_2') = !(1,2)$

at process 1

local choice

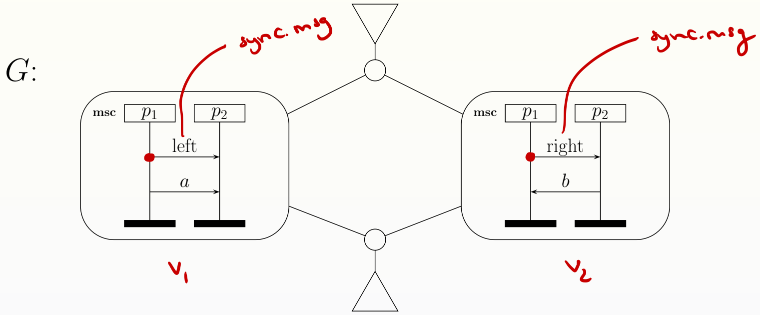
b. v_3: $\pi_3 = v_3 v_3 \dots$ $\min(\pi_3') = !(2,4)$

$\pi_4 = v_3 v_0 \dots$ $\min(\pi_4') = !(2,3)$

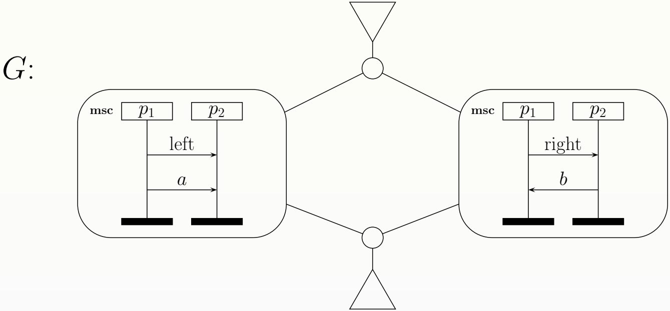
at process 2

local choice

Local choice



Local choice



How to resolve a non-local choice?

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

synchronisation data

Overview

Regular expressions over MSCs

Definition (Asynchronous iteration)

For $\mathcal{M}_1, \mathcal{M}_2 \subseteq \mathbb{M}$ sets of MSCs, let:

$$\mathcal{M}_1 \bullet \mathcal{M}_2 \;=\; \{\, M_1 \bullet M_2 \mid M_1 \in \mathcal{M}_1, M_2 \in \mathcal{M}_2 \,\}$$

For $\mathcal{M} \subseteq \mathbb{M}$ let

$$\mathcal{M}^i \;=\; \begin{cases} \{M_\epsilon\} & \text{if } i{=}0\text{, where } M_\epsilon \text{ denotes the empty MSC} \\ \mathcal{M} \bullet \mathcal{M}^{i-1} & \text{if } i > 0 \end{cases}$$

The asynchronous iteration of $\mathcal{M}$ is now defined by:

$$\mathcal{M}^* = \bigcup_{i \geqslant 0} \mathcal{M}^i.$$

Regular expressions over MSCs

Definition (Regular expressions over MSCs)

The set $\text{REX}_{\mathbb{M}}$ of regular expressions over $\mathbb{M}$ is given by the grammar:

$$\alpha ::= \varnothing \mid M \mid \alpha_1 \cdot \alpha_2 \mid \alpha_1 + \alpha_2 \mid \alpha^*$$

where MSC $M \in \mathbb{M}$.

↳ branching vertex

regular expression

Definition (Semantics of regular expressions, $\mathcal{L}(.) : \text{REX}_{\mathbb{M}} \to 2^{\mathbb{M}}$)

set of MSCs

- ✓ $\mathcal{L}(\varnothing) = \varnothing$ ← empty set of MSCs
- ✓ $\mathcal{L}(M) = \{ M \}$
- ✓ $\mathcal{L}(\alpha_1 \cdot \alpha_2) = \mathcal{L}(\alpha_1) \bullet \mathcal{L}(\alpha_2)$ ← lifting of • to sets of MSCs (cf. previous slide)
- ✓ $\mathcal{L}(\alpha_1 + \alpha_2) = \mathcal{L}(\alpha_1) \cup \mathcal{L}(\alpha_2)$
- ✓ $\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^*$ ← asynchronous on M (cf. previous slide)

Locally accepting CFMs

Definition (Locally accepting CFM)

CFM $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ is locally accepting (la, for short) if

$$F \;=\; \prod_{p \in \mathcal{P}} F_p \quad \text{where} \quad F_p \subseteq S_p.$$

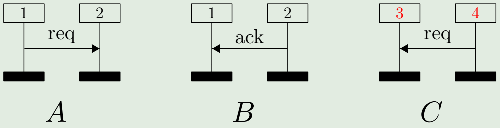
same as for weak CFMs, but now $|\mathbb{D}| > 1$.

Thus: every combination of local accept states is a global accept state of the CFM.

Regular expressions for MSCs

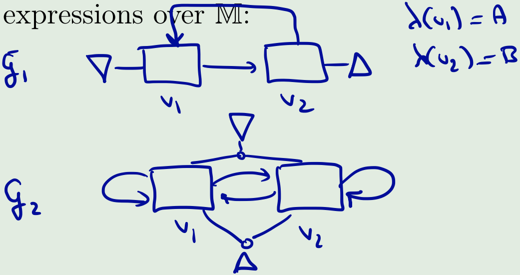
Let $\mathcal{P} = \{1, 2, 3, 4\}$ and $\mathcal{C} = \{\text{req}, \text{ack}\}$.

Example



Consider the following regular expressions over $\mathbb{M}$:

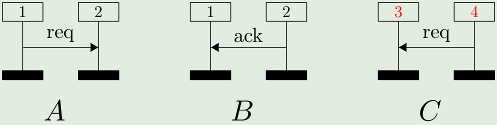
- $\alpha_1 = (A \cdot B)^*$
- $\alpha_2 = (A + B)^*$
- $\alpha_3 = (A \cdot C)^*$
- $\alpha_4 = A \cdot (A + B)^*$



Regular expressions for MSCs

Let $\mathcal{P} = \{1, 2, 3, 4\}$ and $\mathcal{C} = \{\text{req}, \text{ack}\}$.

Example



Consider the following regular expressions over $\mathbb{M}$:

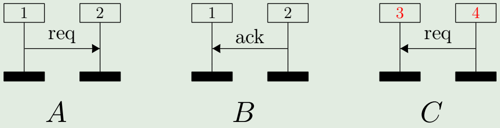
- $\alpha_1 = (A \cdot B)^*$
- $\alpha_2 = (A + B)^*$
- $\alpha_3 = (A \cdot C)^*$
- $\alpha_4 = A \cdot (A + B)^*$

How about realisability of $\mathcal{L}(\alpha_i)$?

Regular expressions for MSCs

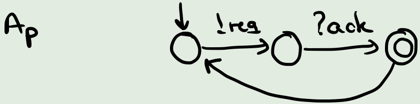
Let $\mathcal{P} = \{1, 2, 3, 4\}$ and $\mathcal{C} = \{\text{req}, \text{ack}\}$.

Example



Consider the following regular expressions over $\mathbb{M}$:

- $\alpha_1 = (A \cdot B)^*$ det. ∀1-bounded dl-free weak CFM
- $\alpha_2 = (A + B)^*$
- $\alpha_3 = (A \cdot C)^*$
- $\alpha_4 = A \cdot (A + B)^*$

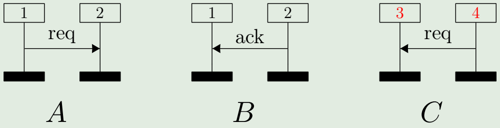


How about realisability of $\mathcal{L}(\alpha_i)$?

Regular expressions for MSCs

Let $\mathcal{P} = \{1, 2, 3, 4\}$ and $\mathcal{C} = \{\text{req}, \text{ack}\}$.

Example



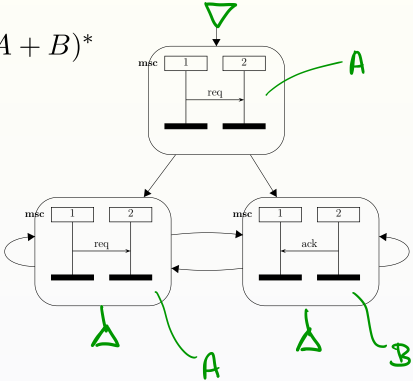
Consider the following regular expressions over $\mathbb{M}$:

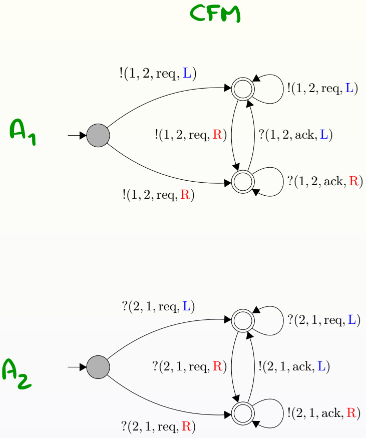
- $\alpha_1 = (A \cdot B)^*$ det. ∀1-bounded dl-free weak CFM
- $\alpha_2 = (A + B)^*$ det. ∃1-bounded la CFM
- $\alpha_3 = (A \cdot C)^*$ not realisable
- → $\alpha_4 = A \cdot (A + B)^*$ ∃1-bounded dl-free locally accepting CFM

How about realisability of $\mathcal{L}(\alpha_i)$?

Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$



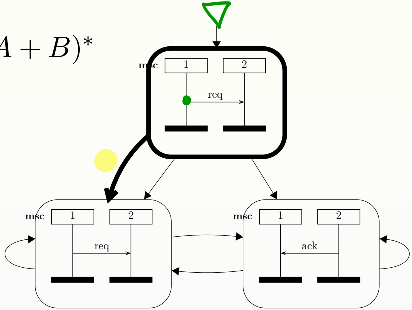


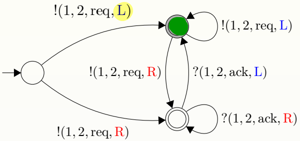
FIFO queues

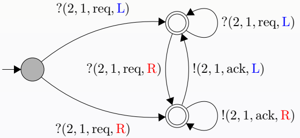
$1 \rightarrow 2$:	empty
$2 \rightarrow 1$:	

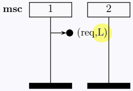
Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$





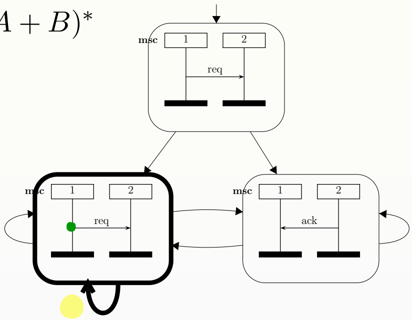


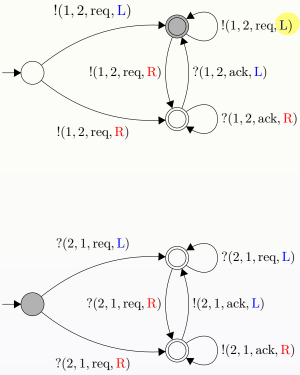


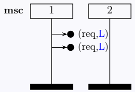
$1 \rightarrow 2$: (req, L)
$2 \rightarrow 1$:

Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$



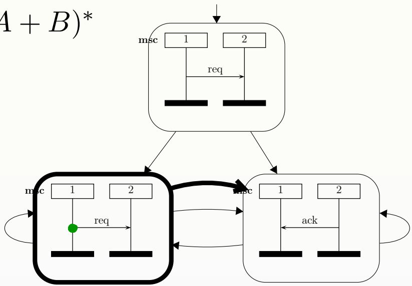


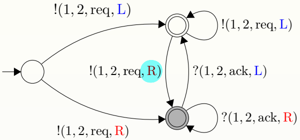


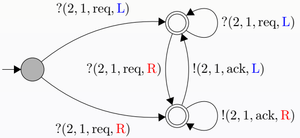
$1 \rightarrow 2$: (req,L) (req,L)
$2 \rightarrow 1$:

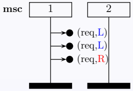
Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$





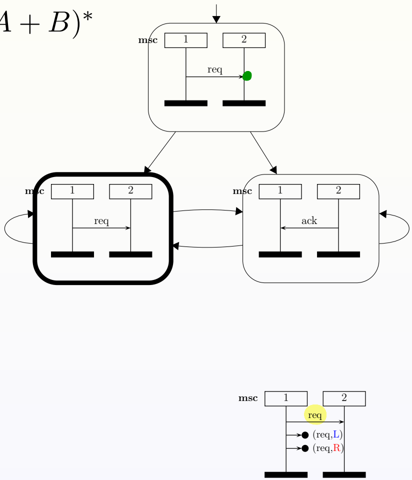


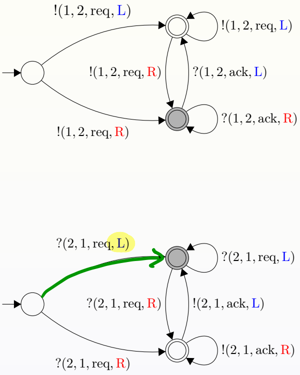


$1 \to 2$: (req,L) (req,L) (req,R)
$2 \to 1$:

Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$

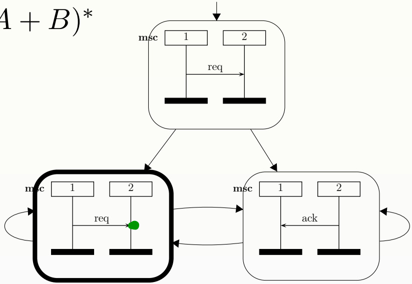


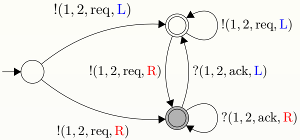


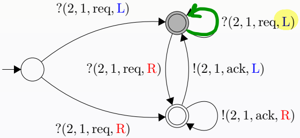
$1 \rightarrow 2$: (req,L) (req,R)
$2 \rightarrow 1$:

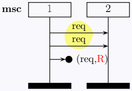
Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$





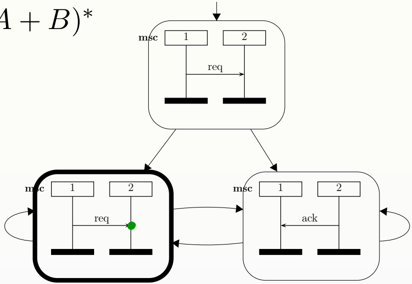


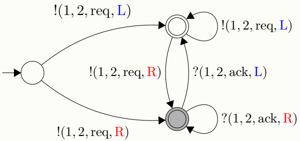


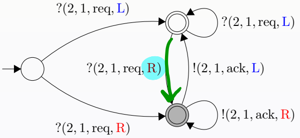
$1 \rightarrow 2$: (req, R)
$2 \rightarrow 1$:

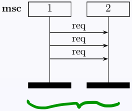
Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$





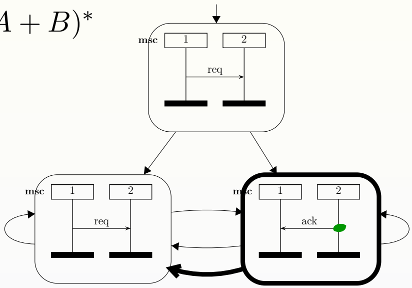


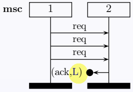


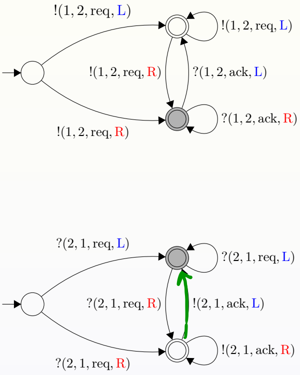
$1 \to 2$:
$2 \to 1$: empty

Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$



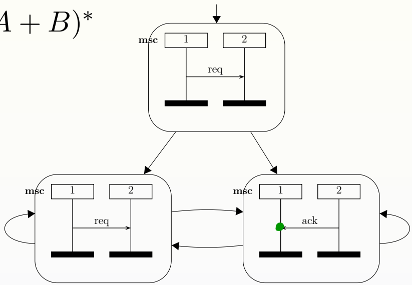


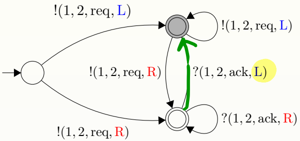


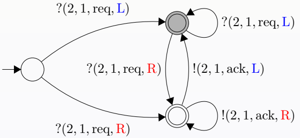
$1 \rightarrow 2$:
$2 \rightarrow 1$: (ack,L)

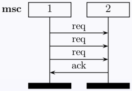
Realising local-choice expressions by deadlock-free CFMs

$A \cdot (A + B)^*$









$1 \rightarrow 2$:
$2 \rightarrow 1$:

Star-connected regular expressions

Definition (Connected MSC)

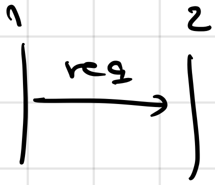
An MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, <) \in \mathbb{M}$ is connected if its communication graph is strongly connected.

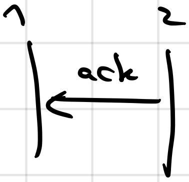
Definition (Star-connected)

Regular expression $\alpha \in \text{REX}_{\mathbb{M}}$ is star-connected if, for any subexpression β^* of α, $\mathcal{L}(\beta)$ is a set of connected MSCs.

Examples on the black board.

(1) $\alpha_1 = (\underbrace{A \cdot B}_{\beta})^*$

$A =$ 

$B =$ 

$\beta = A \cdot B$

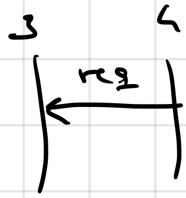
$L(A \cdot B)$ 

connected

$\Rightarrow$ α_1 is star-connected.

(2) $\alpha_3 = (\underbrace{A \cdot C}_{\beta})^*$

$A =$ 

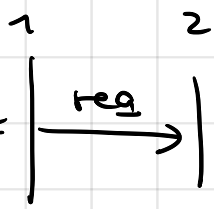
$C =$ 

$L(A \cdot C)$ 

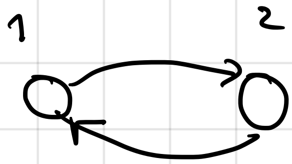
not strongly connected

$\Rightarrow$ α_3 is not star-connected.

(3) $\alpha_4 = A \cdot (\underbrace{A + B}_{\beta})^*$

$A =$ 

$B =$ 

$L(A + B)$ 

$\Rightarrow$ α_4 is star connected.

Regular expressions vs. CFMs

Definition (Finitely generated)

Set of MSCs $\mathcal{M} \subseteq \mathbb{M}$ is finitely generated if there is a finite set of MSCs $\widehat{\mathcal{M}} \subseteq \mathbb{M}$ such that $\mathcal{M} \subseteq \widehat{\mathcal{M}}^*$.

Theorem [Morin 2002]

Let $\mathcal{M}$ be finitely generated. Then:

$\mathcal{M}$ is regular (thus realisable)

iff

there exists a star-connected regular expression α with $\mathcal{L}(\alpha) = \mathcal{M}$.