Local testing of MSCs

Paul Gastin

LSV, ENS Cachan

Joint work with
Puneet Bhateja, Madhavan Mukund, K Narayan Kumar
CMI, Chennai

ANR DOTS, Bordeaux
31 January 2008

Plan

1. MSC

HMSC

Local testing of HMSC

Undecidability of 1-testability for 2 processes

Undecidability of k-testability

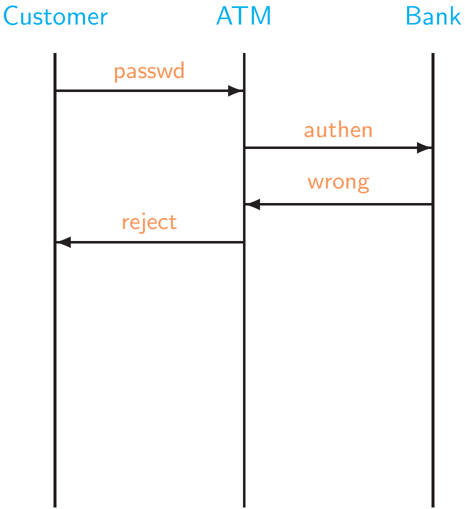
Decidability of 1-testability without message contents

Scenarios

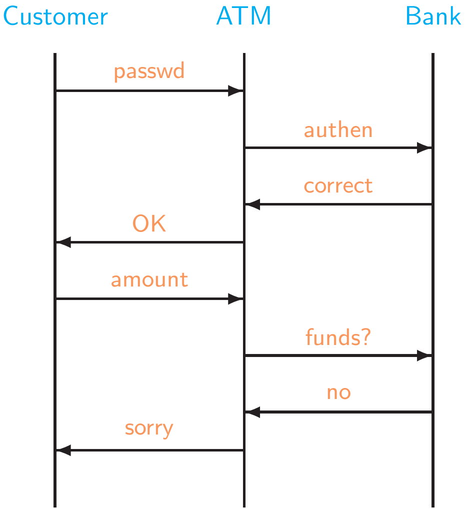
- A scenario describes a pattern of interaction
- Attractive visual formalism
- Telecommunications
  - Message sequence charts (MSC)
  - Messages sent between communicating agents
- UML
  - Sequence diagrams
  - Interaction between objects

    e.g., method invocations etc

An ATM

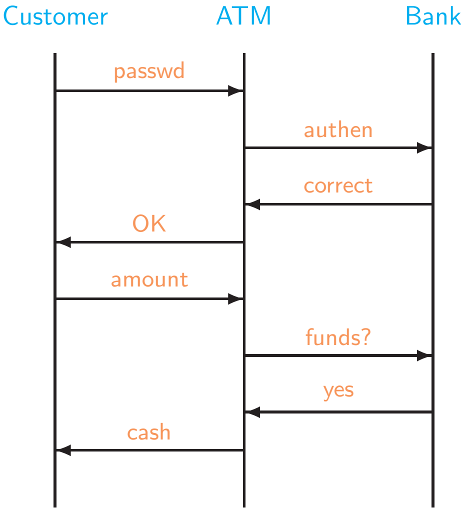


An ATM

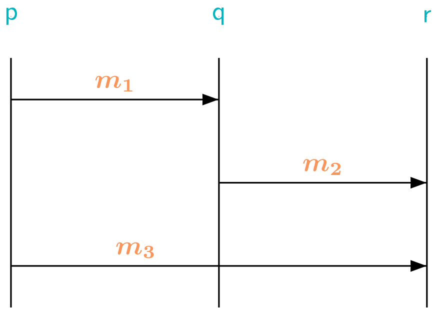


Customer
ATM
Bank
passwd
authen
correct
OK
amount
funds?
no
sorry

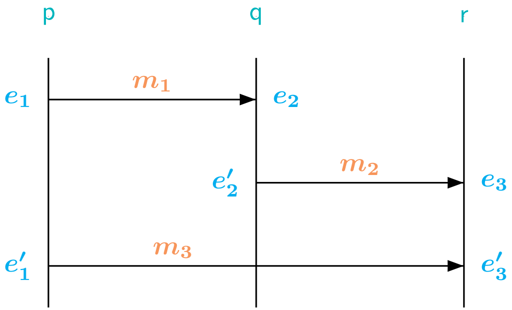
An ATM


Customer
ATM
Bank
passwd
authen
correct
OK
amount
funds?
yes
cash

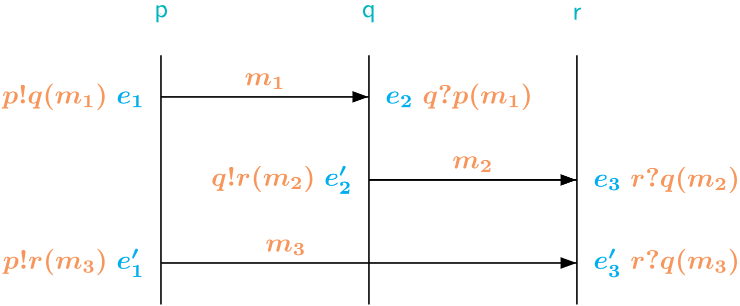
How do we formalize MSCs?



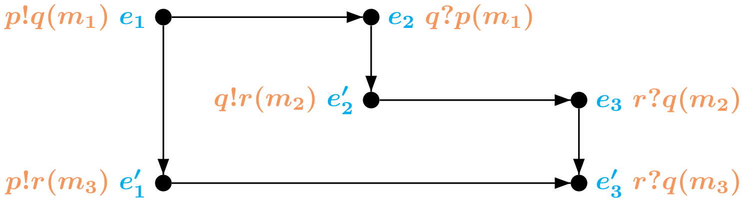
An MSC with events



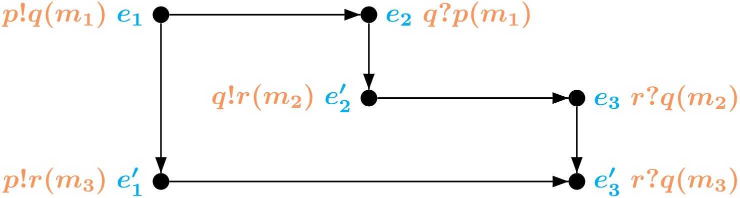
An MSC with labelled events



MSCs as labelled partial orders

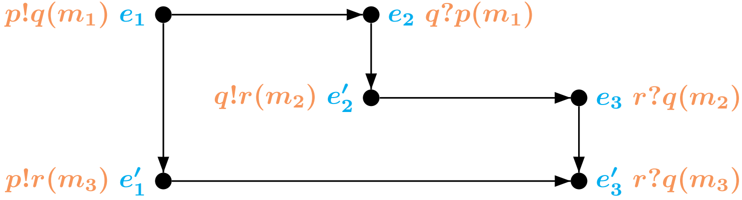


MSCs as labelled partial orders



- Linearizations give a word language
  $p!q(m_1)\ p!r(m_3)\ q?p(m_1)\ q!r(m_2)\ r?q(m_2)\ r?q(m_3)$,
  $p!q(m_1)\ q?p(m_1)\ q!r(m_2)\ p!r(m_3)\ r?q(m_2)\ r?q(m_3)$,...

MSCs as labelled partial orders



- Linearizations give a word language
  $p!q(m_1)\ p!r(m_3)\ q?p(m_1)\ q!r(m_2)\ r?q(m_2)\ r?q(m_3)$,
  $p!q(m_1)\ q?p(m_1)\ q!r(m_2)\ p!r(m_3)\ r?q(m_2)\ r?q(m_3)$,...
- A single linearization is sufficient to reconstruct MSC

Plan

1. MSC
2. HMSC
3. Local testing of HMSC
4. Undecidability of 1-testability for 2 processes
5. Undecidability of k-testability
6. Decidability of 1-testability without message contents

Collections of MSCs

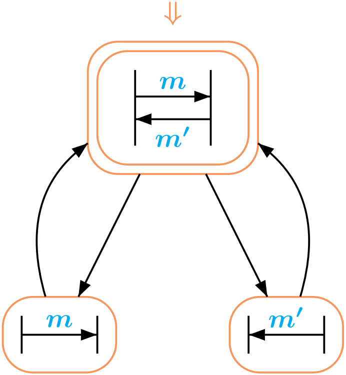
- Often need to specify a collection of scenarios
- Finite collection can be exhaustively enumerated
- Infinite collection needs a generating mechanism

High level MSCs (HMSCs)

- A finite state automaton
- Each state is labelled by an MSC
- Each (legal) path in the automaton generates an MSC

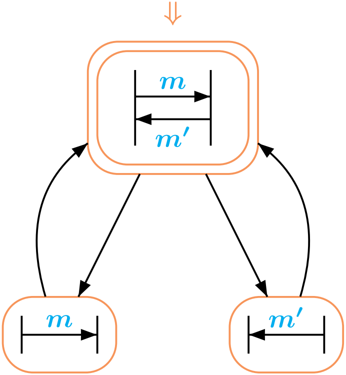
High level MSCs (HMSCs)

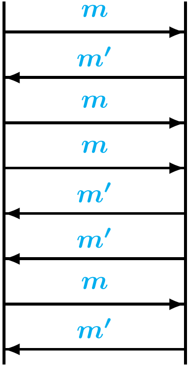
- A finite state automaton
- Each state is labelled by an MSC
- Each (legal) path in the automaton generates an MSC



High level MSCs (HMSCs)

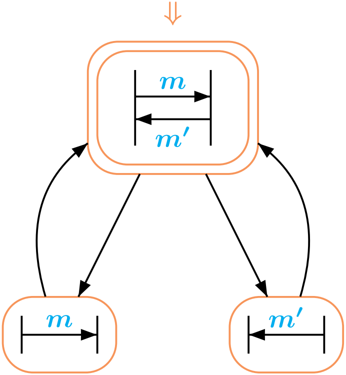
- A finite state automaton
- Each state is labelled by an MSC
- Each (legal) path in the automaton generates an MSC

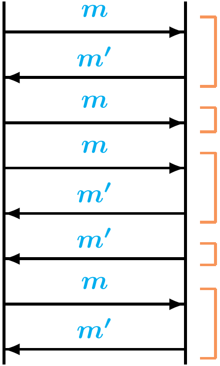




High level MSCs (HMSCs)

- A finite state automaton
- Each state is labelled by an MSC
- Each (legal) path in the automaton generates an MSC





HMSC semantics

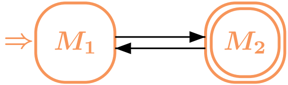
- All processes must traverse the same path in an HMSC

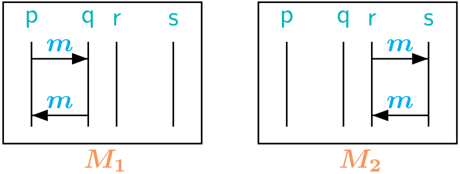
HMSC semantics

- All processes must traverse the same path in an HMSC
- . . . but processes move asynchronously
- Some processes may be (unboundedly) far ahead of others

HMSC semantics

- All processes must traverse the same path in an HMSC
- . . . but processes move asynchronously
- Some processes may be (unboundedly) far ahead of others





- After k iterations, we could have r and s in the final copy of M_2 while p and q are in the first copy of M_1

Regular MSC languages

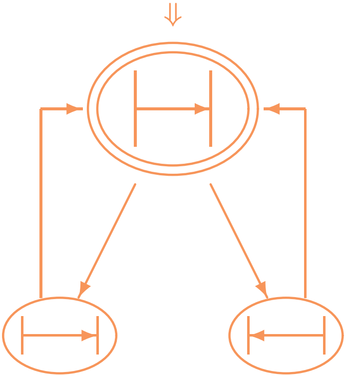
- An MSC is (uniquely) determined by its linearizations
  - Set of strings over send actions $p!q(m)$ and receive actions $p?q(m)$
- Collection of MSCs $\Leftrightarrow$
  word language over send/receive actions
- Regular collection of MSCs $\triangleq$
  linearizations form a regular language

HMSCs and regularity

- HMSC specifications may not be regular

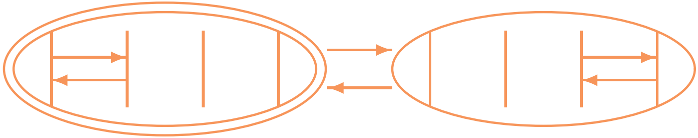
HMSCs and regularity

- HMSC specifications may not be regular
- Problem 1 Unbounded buffers



HMSCs and regularity

- HMSC specifications may not be regular
- Problem 1 Unbounded buffers
- Problem 2 Global synchronization yields context-free behaviours

$\Rightarrow$ 

HMSCs and regularity

- HMSC specifications may not be regular
- Problem 1 Unbounded buffers
- Problem 2 Global synchronization yields context-free behaviours
- Sufficient structural conditions on HMSCs to guarantee regularity . . . [AY99,MP99]
  - Locally synchronized

HMSCs and regularity

- HMSC specifications may not be regular
- Problem 1 Unbounded buffers
- Problem 2 Global synchronization yields context-free behaviours
- Sufficient structural conditions on HMSCs to guarantee regularity ... [AY99,MP99]
  - Locally synchronized
- ... but checking if an HMSC specification is regular is undecidable [HMNST05]

HMSCs and regularity

- HMSC specifications may not be regular
- Problem 1 Unbounded buffers
- Problem 2 Global synchronization yields context-free behaviours
- Sufficient structural conditions on HMSCs to guarantee regularity . . . [AY99,MP99]
  - Locally synchronized
- . . . but checking if an HMSC specification is regular is undecidable [HMNST05]
- Every regular MSC language can be implemented as network of communicating finite-state automata with bounded channels [HMNST05]

Plan

1. MSC
2. HMSC
3. Local testing of HMSC
4. Undecidability of 1-testability for 2 processes
5. Undecidability of k-testability
6. Decidability of 1-testability without message contents

(Local) testing using scenarios

- Does an implementation conform to an HMSC specification?

(Local) testing using scenarios

- Does an implementation conform to an HMSC specification?
- Local testing

(Local) testing using scenarios

- Does an implementation conform to an HMSC specification?
- Local testing
  - Inject messages from some process(es) and observe the response

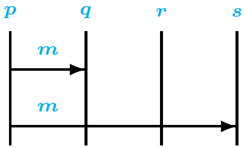
(Local) testing using scenarios

- Does an implementation conform to an HMSC specification?
- Local testing
  - Inject messages from some process(es) and observe the response
  - For each process p, local observer records sequence of events at p
  - If each local observation is consistent with some MSC defined by the HMSC, the implementation passes the test

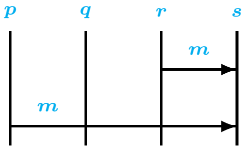
(Local) testing using scenarios

- Does an implementation conform to an HMSC specification?
- Local testing
  - Inject messages from some process(es) and observe the response
  - For each process p, local observer records sequence of events at p
  - If each local observation is consistent with some MSC defined by the HMSC, the implementation passes the test
- Does local testing suffice to check conformance of (regular) HMSC languages?

Implied scenarios [AEY00]

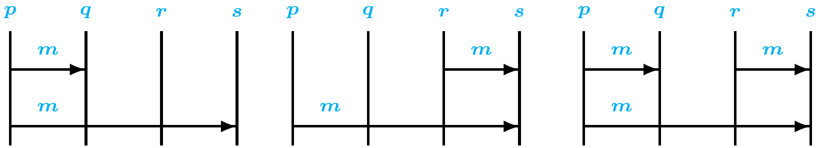


M_1



M_2

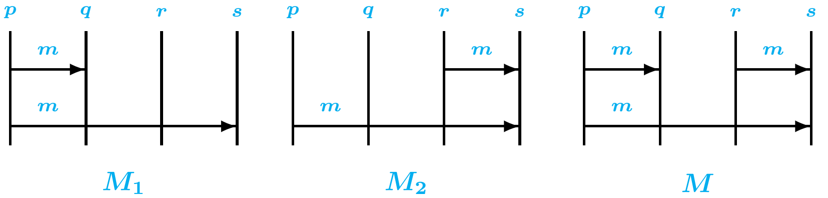
Implied scenarios [AEY00]



M_1 M_2 M

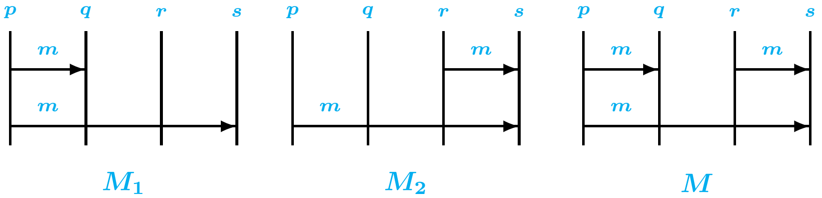
- p and q believe M is M_1
- r and s believe M is M_2

Implied scenarios [AEY00]



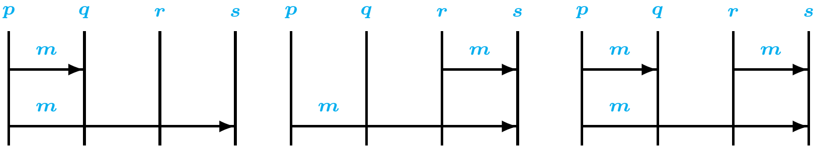
- p and q believe M is M_1
- r and s believe M is M_2
- MSC M is implied by L if for each process p, the p-projection of M matches the p-projection of some MSC in L

Implied scenarios [AEY00]



- p and q believe M is M_1
- r and s believe M is M_2
- MSC M is implied by L if for each process p, the p-projection of M matches the p-projection of some MSC in L
- An MSC language is locally testable if it is closed with respect to implied MSCs

Implied scenarios [AEY00]



M_1 M_2 M

- p and q believe M is M_1
- r and s believe M is M_2
- MSC M is implied by L if for each process p, the p-projection of M matches the p-projection of some MSC in L
- An MSC language is locally testable if it is closed with respect to implied MSCs
- Originally studied in context of realizability

Implied scenarios . . .

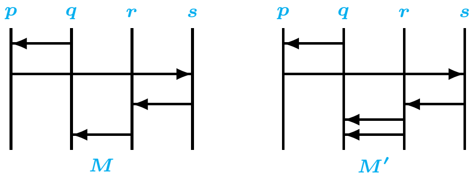
- Even for regular MSC languages, checking local testability is undecidable! [AEY01]

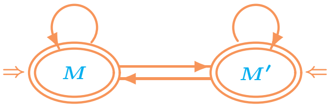
Implied scenarios . . .

- Even for regular MSC languages, checking local testability is undecidable! [AEY01]
- Even if the original language has bounded channels, its implied scenarios may not

Implied scenarios . . .

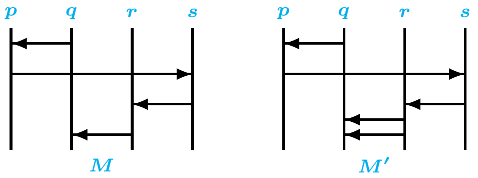
- Even for regular MSC languages, checking local testability is undecidable! [AEY01]
- Even if the original language has bounded channels, its implied scenarios may not

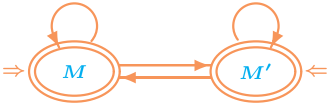


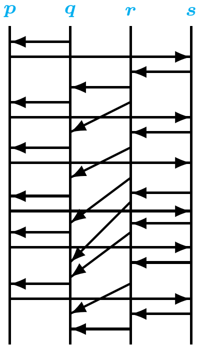


Implied scenarios . . .

- Even for regular MSC languages, checking local testability is undecidable! [AEY01]
- Even if the original language has bounded channels, its implied scenarios may not

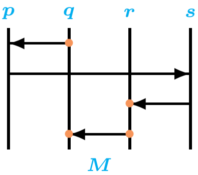


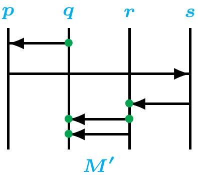


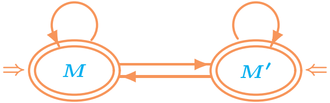


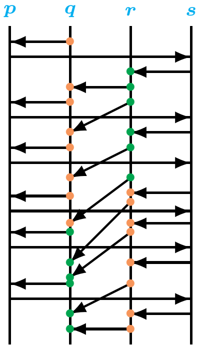
Implied scenarios . . .

- Even for regular MSC languages, checking local testability is undecidable! [AEY01]
- Even if the original language has bounded channels, its implied scenarios may not



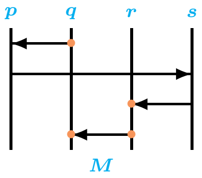


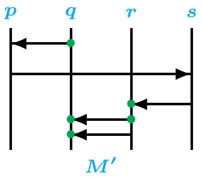




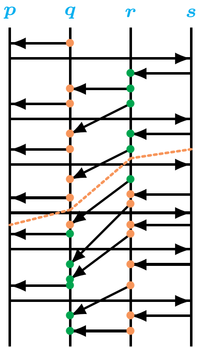
Implied scenarios . . .

- Even for regular MSC languages, checking local testability is undecidable! [AEY01]
- Even if the original language has bounded channels, its implied scenarios may not





Confusing $M^{2k}M'^{k}$ and $M'^{k}M^{2k}$ generates upto k messages in $p \longrightarrow s$ channel

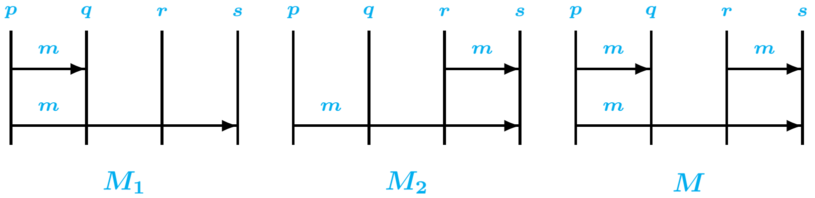


Joint observations

- What if we have observers who can record the behaviours of sets of processes?

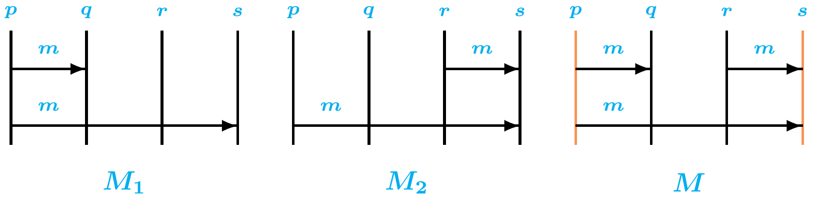
Joint observations

- What if we have observers who can record the behaviours of sets of processes?



Joint observations

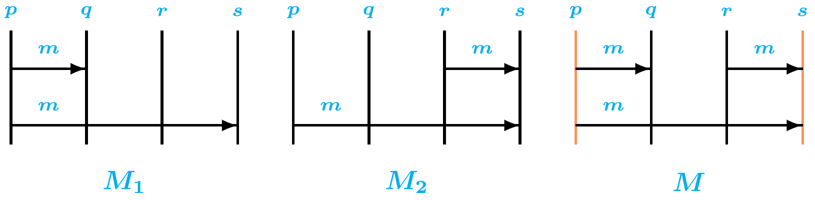
- What if we have observers who can record the behaviours of sets of processes?



- M is detected as an illegal MSC by $\{p, s\}$.

Joint observations

- What if we have observers who can record the behaviours of sets of processes?



- M is detected as an illegal MSC by $\{p, s\}$.
- Joint observers have more discriminating power.

Joint observations . . .

- Fix some *observers* $P_1, P_2, \ldots, P_r$
- Each observer records the events on the processes in the set P_i

Joint observations . . .

- Fix some *observers* $P_1, P_2, \ldots, P_r$
- Each observer records the events on the processes in the set P_i

Given a HMSC G, is its language testable with observers $P_1, P_2, \ldots, P_r$?

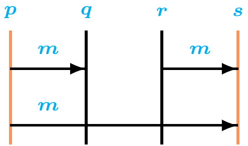
P-Observations

M an MSC, P a set of processes

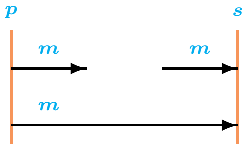
P-observation of $M \stackrel{\triangle}{=}$ tuple of projections of M on each process in P

$M\restriction_P$: P-observation of M.

$L\restriction_P = \{M\restriction_P \mid M \in L\}$: P-observation of a language L



M



$M\restriction_{p,s}$

$$M\restriction_{p,s} = \langle p!q(m)p!s(m), s?r(m)s?p(m)\rangle.$$

k-testability

- Record P-observations for every set P of processes of size k.

k-testability

- Record P-observations for every set P of processes of size k.
- k-closure of a language $L \triangleq \{M \mid \forall P \text{ s.t. } |P| = k,\ M\restriction_P \in L\restriction_P\}$

k-testability

- Record P-observations for every set P of processes of size k.
- k-closure of a language $L \triangleq \{M \mid \forall P \text{ s.t. } |P| = k,\ M{\restriction_P} \in L{\restriction_P}\}$
- Scenario k-implied by $L \triangleq$ MSC in the k-closure of L but not in L

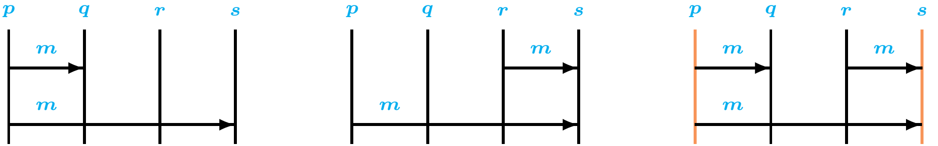
k-testability

- Record P-observations for every set P of processes of size k.
- k-closure of a language $L \triangleq \{M \mid \forall P \text{ s.t. } |P| = k,\ M\restriction_P \in L\restriction_P\}$
- Scenario k-implied by $L \triangleq$ MSC in the k-closure of L but not in L
- A language is k-testable if it equals its k-closure

k-testability

- Record P-observations for every set P of processes of size k.
- k-closure of a language $L \triangleq \{M \mid \forall P \text{ s.t. } |P| = k,\ M\restriction_P \in L\restriction_P\}$
- Scenario k-implied by $L \triangleq$ MSC in the k-closure of L but not in L
- A language is k-testable if it equals its k-closure
- Local testability is 1-testability

k-testability ...



M_1 M_2 M

The set $\{M_1, M_2\}$ is 2-testable but not 1-testable.

k-testability . . .

- 1-testability is undecidable for 4 or more processes. [AEY 01]

k-testability . . .

- 1-testability is undecidable for 4 or more processes. [AEY 01]
- n-testability is trivial

k-testability . . .

- 1-testability is undecidable for 4 or more processes. [AEY 01]
- n-testability is trivial
- What about k-testability for $1 < k < n$?

k-testability ...

- 1-testability is undecidable for 4 or more processes. [AEY 01]
- n-testability is trivial
- What about k-testability for $1 < k < n$?
- What is the smallest $k \leq n$ such that k-testability is decidable?

Our results

- For all n and $k < n$ there are regular HMSC languages over n processes that are not k-testable

Our results

- For all n and $k < n$ there are regular HMSC languages over n processes that are not k-testable
- k-testability is undecidable for $n \geq 3$ processes and $1 < k < n$

Our results

- For all n and $k < n$ there are regular HMSC languages over n processes that are not k-testable
- k-testability is undecidable for $n \geq 3$ processes and $1 < k < n$
- 1-testability is undecidable for 2 processes

Our results

- For all n and $k < n$ there are regular HMSC languages over n processes that are not k-testable
- k-testability is undecidable for $n \geq 3$ processes and $1 < k < n$
- 1-testability is undecidable for 2 processes
  - Improves result from 4 processes in [AEY01]

Our results

- For all n and $k < n$ there are regular HMSC languages over n processes that are not k-testable
- k-testability is undecidable for $n \geq 3$ processes and $1 < k < n$
- 1-testability is undecidable for 2 processes
  - Improves result from 4 processes in [AEY01]
- k-testability remains undecidable for $n \geq 3$ processes and $1 < k < n$ even without message contents

Our results

- For all n and $k < n$ there are regular HMSC languages over n processes that are not k-testable
- k-testability is undecidable for $n \geq 3$ processes and $1 < k < n$
- 1-testability is undecidable for 2 processes
  - Improves result from 4 processes in [AEY01]
- k-testability remains undecidable for $n \geq 3$ processes and $1 < k < n$ even without message contents
- 1-testability is decidable without message contents

Plan

MSC

HMSC

Local testing of HMSC

4. Undecidability of 1-testability for 2 processes

Undecidability of k-testability

Decidability of 1-testability without message contents

1-testability for 2 processes

Theorem : 2 processes

For $n \geq 2$, 1-testability is undecidable for regular 4-bounded MSG-definable languages over n processes.

Proof :

Reduction to Modified Post Correspondence Problem (MPCP).

Modified Post Correspondence Problem

Definition : MPCP

Instance: sequence $(v_1, w_1), (v_2, w_2), \ldots, (v_r, w_r)$ of pairs of words such that

- $1 \leq |v_i| \leq 4$ and $1 \leq |w_i| \leq 4$ for $1 \leq i \leq r$,
- $w_1 < v_1$ and is shorter by at least 2 letters.

Solution: sequence $1 = i_1, i_2, i_3, \ldots, i_m$ of indices from $\{1, 2, \ldots, r\}$ such that

$$w_{i_1} w_{i_2} \cdots w_{i_m} = v_{i_1} v_{i_2} \cdots v_{i_m}$$

and for $k < m$,

$$w_{i_1} w_{i_2} \cdots w_{i_k} < v_{i_1} v_{i_2} \cdots v_{i_k}$$

Theorem : MPCP

The Modified Post Correspondence Problem is undecidable.

Undecidability: Reduction

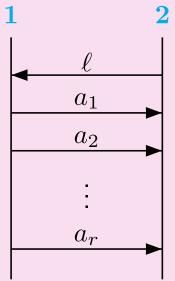
Proof :

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.

Undecidability: Reduction

Proof :

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.

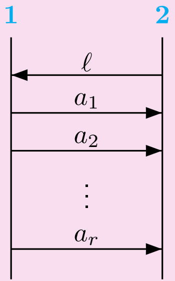


M_{v_ℓ} with $v_\ell = a_1 a_2 \ldots a_r$

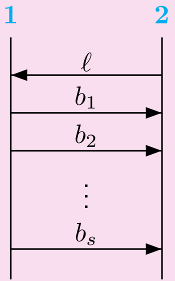
Undecidability: Reduction

Proof :

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



M_{v_ℓ} with $v_\ell = a_1 a_2 \ldots a_r$

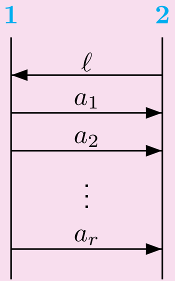


M_{w_ℓ} with $w_\ell = b_1 b_2 \ldots b_s$

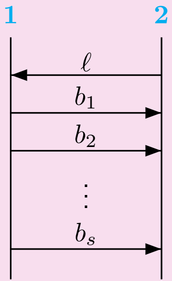
Undecidability: Reduction

Proof :

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



M_{v_ℓ} with $v_\ell = a_1 a_2 \ldots a_r$



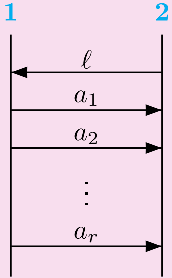
M_{w_ℓ} with $w_\ell = b_1 b_2 \ldots b_s$

$$L_\Delta = M_{v_1}.\{M_{v_1}, \ldots, M_{v_t}\}^* + M_{w_1}.\{M_{w_1}, \ldots, M_{w_t}\}^*$$

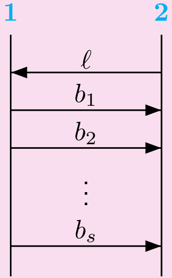
Undecidability: Reduction

Proof :

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



M_{v_ℓ} with $v_\ell = a_1 a_2 \ldots a_r$



M_{w_ℓ} with $w_\ell = b_1 b_2 \ldots b_s$

$$L_\Delta = M_{v_1}.\{M_{v_1}, \ldots, M_{v_t}\}^* + M_{w_1}.\{M_{w_1}, \ldots, M_{w_t}\}^*$$

Lemma :

The MPCP Δ has a solution iff L_Δ has some 1-implied scenario.

Undecidability: Reduction

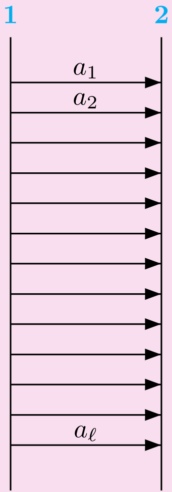
Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$

Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

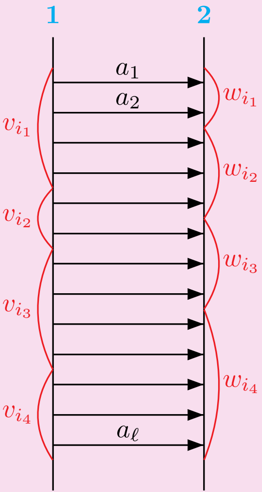
$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

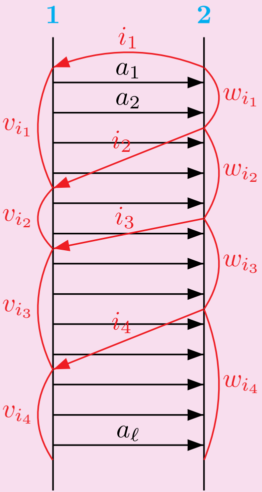
$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

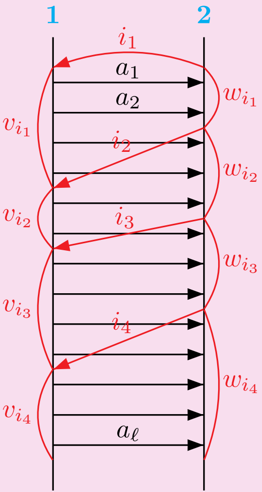
$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



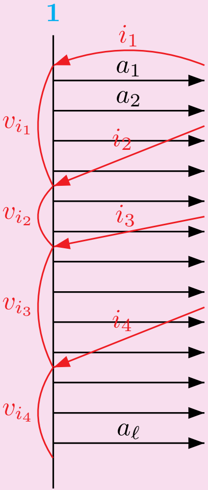
$$L = M_{v_1}.\{M_{v_1}, \ldots, M_{v_t}\}^* + M_{w_1}.\{M_{w_1}, \ldots, M_{w_t}\}^*$$

- $M \notin L$

Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



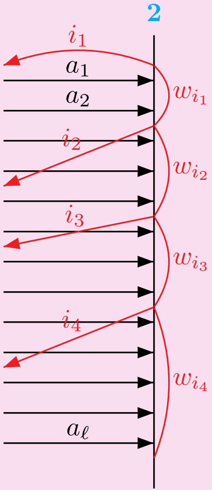
$$L = M_{v_1}.\{M_{v_1}, \ldots, M_{v_t}\}^* + M_{w_1}.\{M_{w_1}, \ldots, M_{w_t}\}^*$$

- $M \notin L$
- $M\restriction_1 = (M_{v_{i_1}} M_{v_{i_2}} \ldots M_{v_{i_m}})\restriction_1 \in L\restriction_1$

Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



$$L = M_{v_1}.\{M_{v_1}, \ldots, M_{v_t}\}^* + M_{w_1}.\{M_{w_1}, \ldots, M_{w_t}\}^*$$

- $M \notin L$
- $M\restriction_1 = (M_{v_{i_1}} M_{v_{i_2}} \ldots M_{v_{i_m}})\restriction_1 \in L\restriction_1$
- $M\restriction_2 = (M_{w_{i_1}} M_{w_{i_2}} \ldots M_{w_{i_m}})\restriction_2 \in L\restriction_2$

Plan

MSC

HMSC

Local testing of HMSC

Undecidability of 1-testability for 2 processes

5 Undecidability of k-testability

Decidability of 1-testability without message contents

k-testability

Theorem : k-testability

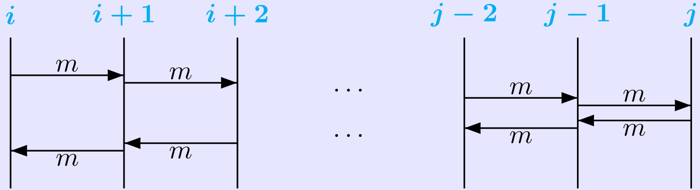
For $1 < k < n$, k-testability is undecidable for regular 1-bounded MSG-definable languages over n processes.

Proof :

Reduction to Modified Post Correspondence Problem (MPCP).

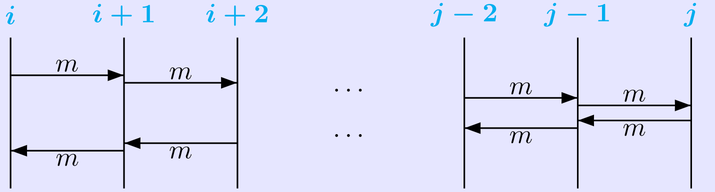
Undecidability: Reduction

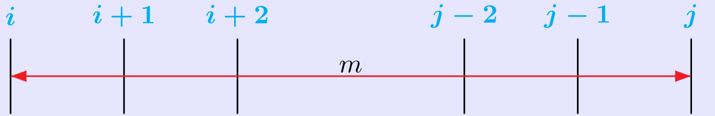
A basic MSC



Undecidability: Reduction

A basic MSC



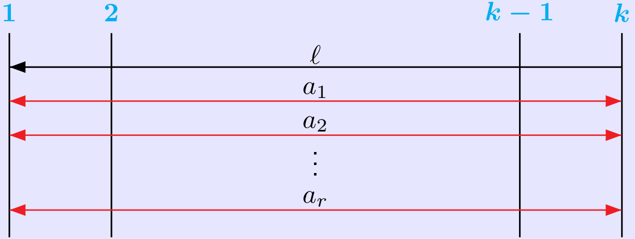


Undecidability: Reduction

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.

Undecidability: Reduction

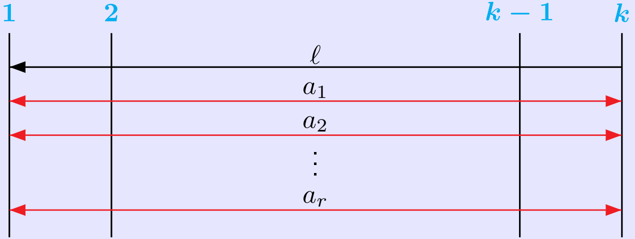
Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



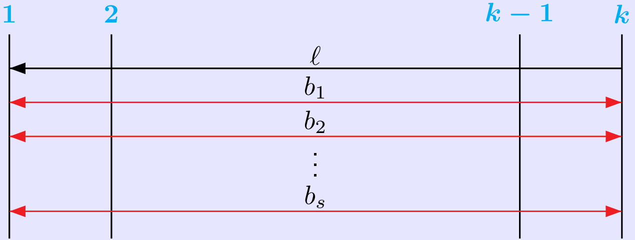
M_{v_ℓ} with
$v_\ell = a_1 a_2 \ldots a_r$

Undecidability: Reduction

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



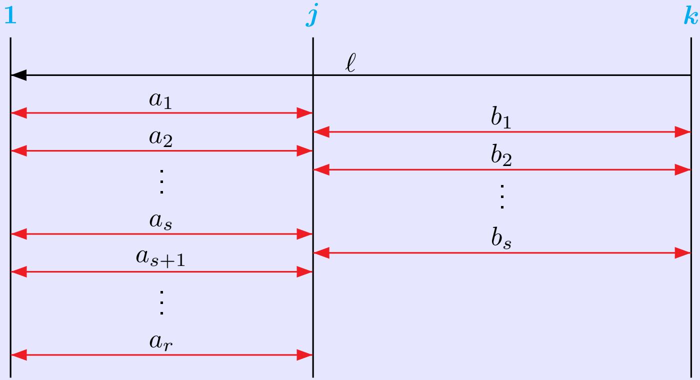
M_{v_ℓ} with
$v_\ell = a_1 a_2 \ldots a_r$



M_{w_ℓ} with
$w_\ell = b_1 b_2 \ldots b_s$

Undecidability: Reduction

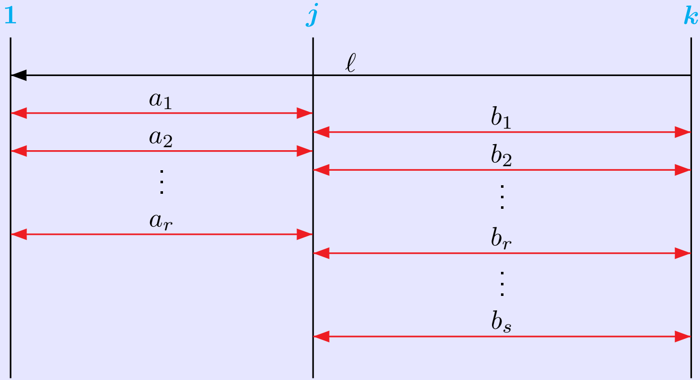
Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



$M^j_{v_\ell, w_\ell}$ with $v_\ell = a_1 a_2 \ldots a_r$ and $w_\ell = b_1 b_2 \ldots b_s$

Undecidability: Reduction

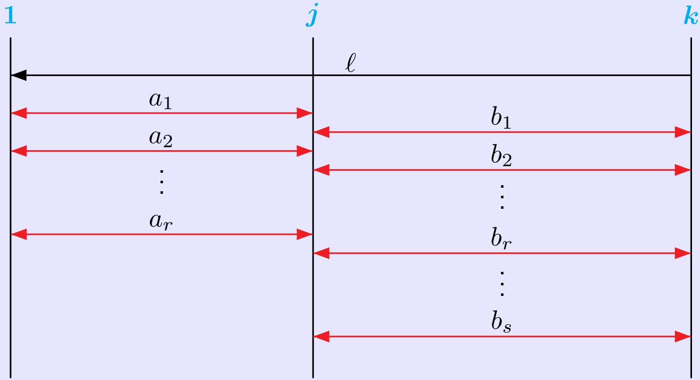
Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



$M^j_{v_\ell, w_\ell}$ with $v_\ell = a_1 a_2 \ldots a_r$ and $w_\ell = b_1 b_2 \ldots b_s$

Undecidability: Reduction

Let $\Delta = (v_1, w_1), (v_2, w_2), \ldots, (v_t, w_t)\}$ be an instance of the MPCP.



$M^j_{v_\ell, w_\ell}$ with $v_\ell = a_1 a_2 \ldots a_r$ and $w_\ell = b_1 b_2 \ldots b_s$

$$L_\Delta = M_{v_1}.\{M_{v_1}, \ldots, M_{v_t}\}^* \cup M_{w_1}.\{M_{w_1}, \ldots, M_{w_t}\}^* \cup \bigcup_j M^j_{v_1,w_1}\{M^j_{v_1,w_1}, \ldots, M^j_{v_t,w_t}\}^*$$

Undecidability: Reduction

Lemma :

The MPCP Δ has a solution iff L_Δ has some $(k-1)$-implied scenario.

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$

We build an MSC $M \notin L_\Delta$ which is $(k-1)$-implied by L_Δ.

Undecidability: Reduction

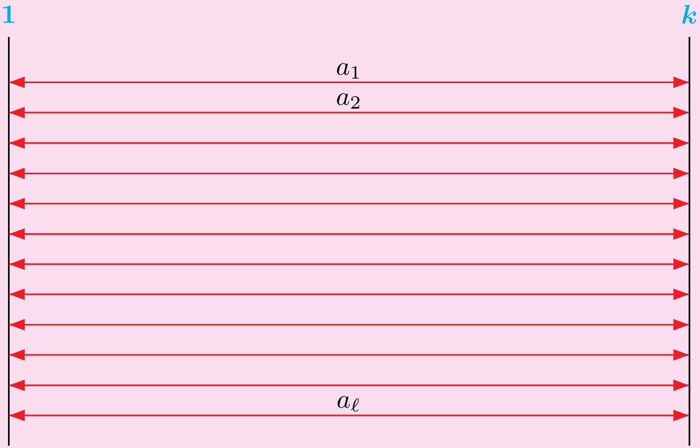
Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$

Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

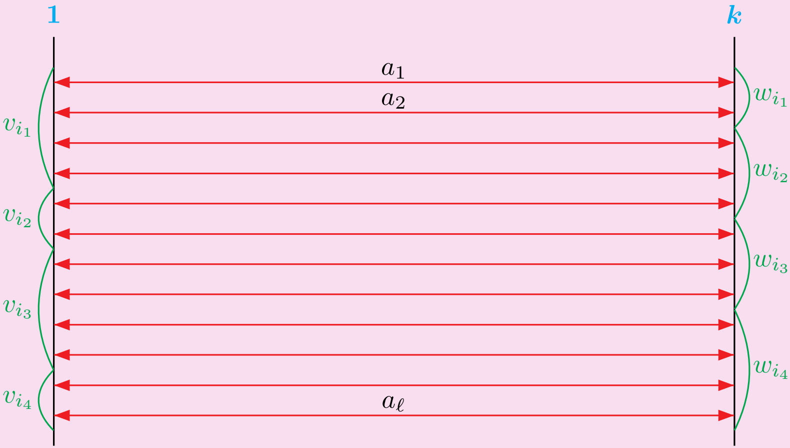
$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

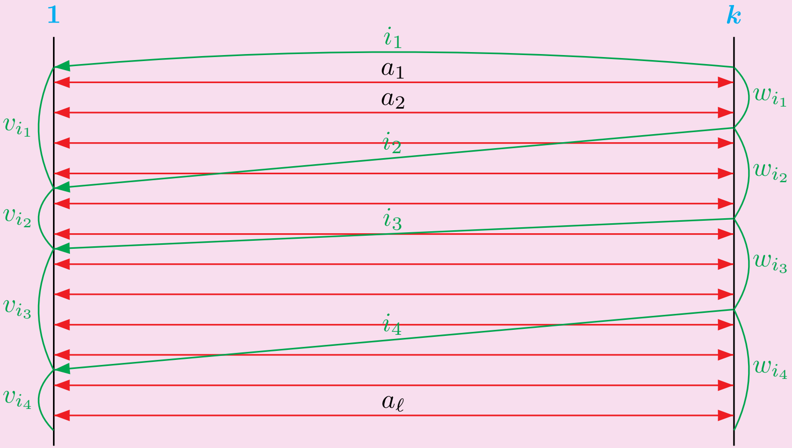
$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$

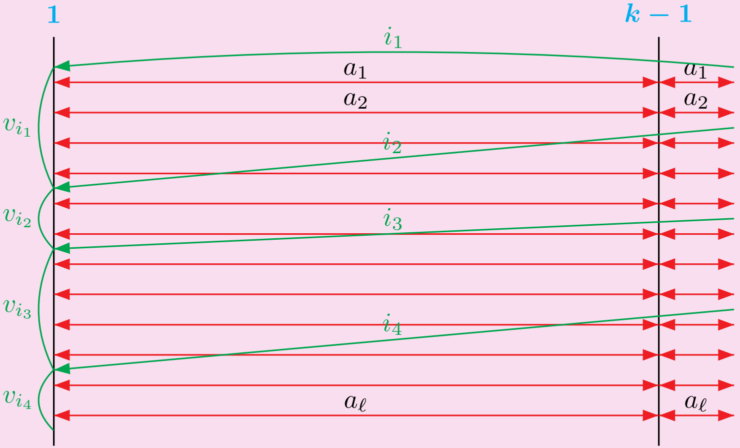


$$M \notin L_\Delta$$

Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$

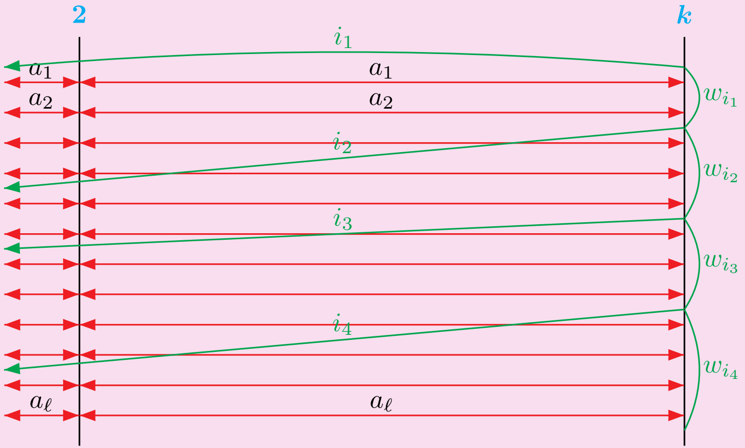


$$M\restriction_{1,\ldots,k-1} = (M_{v_{i_1}} M_{v_{i_2}} \ldots M_{v_{i_m}})\restriction_{1,\ldots,k-1}$$

Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$

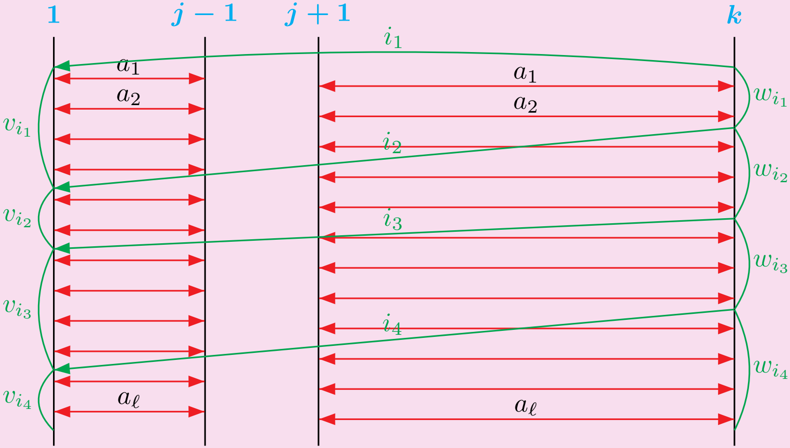


$$M\restriction_{2,\ldots,k} = (M_{w_{i_1}} M_{w_{i_2}} \ldots M_{w_{i_m}})\restriction_{2,\ldots,k}$$

Undecidability: Reduction

Proof : Let $1 = i_1, i_2, \ldots, i_m$ be a solution of MPCP

$$v_{i_1} v_{i_2} \ldots v_{i_m} = a_1 a_2 \ldots a_\ell = w_{i_1} w_{i_2} \ldots w_{i_m}$$



$$M\restriction_{1,\ldots,j-1,j+1,\ldots,k} = (M^j_{v_{i_1},w_{i_1}} M^j_{v_{i_2},w_{i_2}} \ldots M^j_{v_{i_m},w_{i_m}})\restriction_{1,\ldots,j-1,j+1,\ldots,k}$$

Plan

1-testability over the singleton alphabet

- Each channel behaves as counter

1-testability over the singleton alphabet

- Each channel behaves as counter
- Encode the behaviours of the HMSC as a Petri Net

1-testability over the singleton alphabet

- Each channel behaves as counter
- Encode the behaviours of the HMSC as a Petri Net
- HMSC defines a regular language $\Rightarrow$ channels are bounded by some constant B

1-testability over the singleton alphabet

- Each channel behaves as counter
- Encode the behaviours of the HMSC as a Petri Net
- HMSC defines a regular language $\Rightarrow$ channels are bounded by some constant B
    1. Check if net has $B + 1$ messages in a channel en route to final marking

1-testability over the singleton alphabet

- Each channel behaves as counter
- Encode the behaviours of the HMSC as a Petri Net
- HMSC defines a regular language $\Rightarrow$ channels are bounded by some constant B
  1. Check if net has $B+1$ messages in a channel en route to final marking
  2. If yes, implied scenario exists

1-testability over the singleton alphabet

- Each channel behaves as counter
- Encode the behaviours of the HMSC as a Petri Net
- HMSC defines a regular language $\Rightarrow$ channels are bounded by some constant B
  1. Check if net has $B + 1$ messages in a channel en route to final marking
  2. If yes, implied scenario exists
  3. Otherwise, language of net is regular

1-testability over the singleton alphabet

- Each channel behaves as counter
- Encode the behaviours of the HMSC as a Petri Net
- HMSC defines a regular language $\Rightarrow$ channels are bounded by some constant B
  1. Check if net has $B + 1$ messages in a channel en route to final marking
  2. If yes, implied scenario exists
  3. Otherwise, language of net is regular
     - Check if net exhibits any behaviour not described by HMSC

1-testability over the singleton alphabet

- Each channel behaves as counter
- Encode the behaviours of the HMSC as a Petri Net
- HMSC defines a regular language $\Rightarrow$ channels are bounded by some constant B
  1. Check if net has $B + 1$ messages in a channel en route to final marking
  2. If yes, implied scenario exists
  3. Otherwise, language of net is regular
     - Check if net exhibits any behaviour not described by HMSC
- Special case of a result due to Morin [M02]

Future work

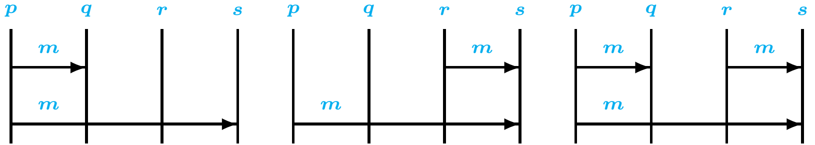
- Local testability is undecidable in most situations

Future work

- Local testability is undecidable in most situations
- Look for sufficient conditions that indicate violation of testability

Future work

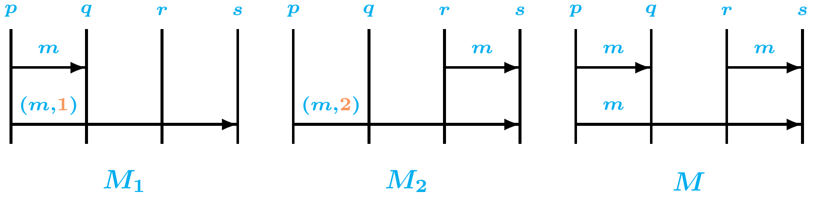
- Local testability is undecidable in most situations
- Look for sufficient conditions that indicate violation of testability
- Testability by piggybacking auxiliary information?



M_1 M_2 M

Future work

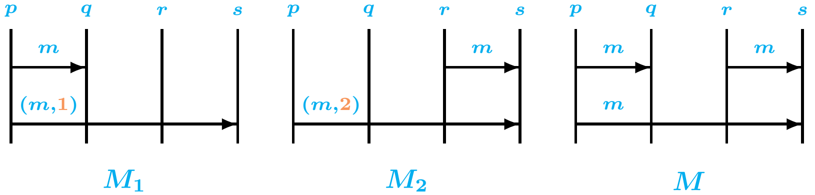
- Local testability is undecidable in most situations
- Look for sufficient conditions that indicate violation of testability
- Testability by piggybacking auxiliary information?



By tagging auxiliary information to m,
p informs s whether it has sent a message to q
This rules out the implied scenario M

Future work

- Local testability is undecidable in most situations
- Look for sufficient conditions that indicate violation of testability
- Testability by piggybacking auxiliary information?

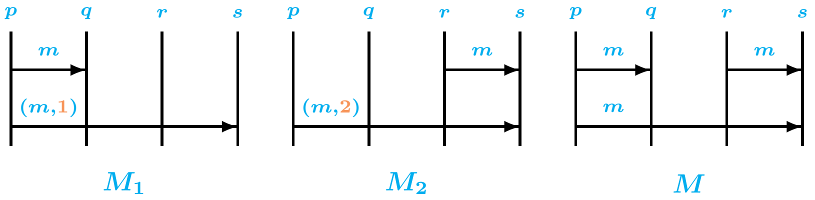


By tagging auxiliary information to m,
p informs s whether it has sent a message to q
This rules out the implied scenario M

- Can we piggyback a bounded amount of auxiliary information to ensure testability?

Future work

- Local testability is undecidable in most situations
- Look for sufficient conditions that indicate violation of testability
- Testability by piggybacking auxiliary information?



  By tagging auxiliary information to m,
  p informs s whether it has sent a message to q
  This rules out the implied scenario M
- Can we piggyback a bounded amount of auxiliary information to ensure testability?
- Bounded auxiliary information suffices to check causal closure [AMNN05]