Analyzing time constrained MSGs

K Narayan Kumar

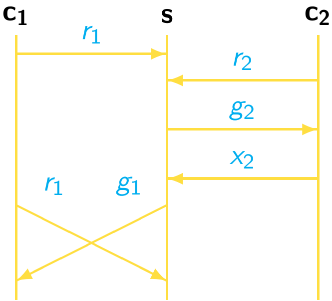
Chennai Mathematical Institute
http://www.cmi.ac.in/~kumar

Joint work with P Gastin, Madhavan Mukund

Chennai, 30 January 2009

Message Sequence Charts

Two clients and a server



Overview

Message Sequence Charts

Overview

Message Sequence Charts

- Visual formalism for specifying scenarios.

Overview

Message Sequence Charts

- Visual formalism for specifying scenarios.
- Part of the UML Standard

Overview

Message Sequence Charts

- Visual formalism for specifying scenarios.
- Part of the UML Standard
- Has a rich and well understood theory.

Overview

Message Sequence Charts

- Visual formalism for specifying scenarios.
- Part of the UML Standard
- Has a rich and well understood theory.

- Timing constraints are natural for scenario specifications

Overview

Message Sequence Charts

- Visual formalism for specifying scenarios.
- Part of the UML Standard
- Has a rich and well understood theory.

- Timing constraints are natural for scenario specifications
- If acknowledgment is not received within a reasonable amount of time, retransmit . . .

Overview

Message Sequence Charts

- Visual formalism for specifying scenarios.
- Part of the UML Standard
- Has a rich and well understood theory.

- Timing constraints are natural for scenario specifications
- If acknowledgment is not received within a reasonable amount of time, retransmit . . .

Can we extend the analysis techniques to the timed setting?

Overview

Message Sequence Charts

- Visual formalism for specifying scenarios.
- Part of the UML Standard
- Has a rich and well understood theory.

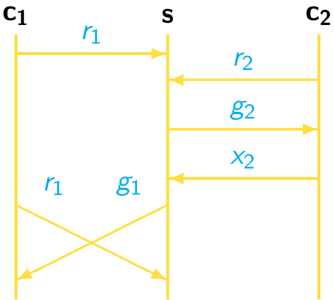
- Timing constraints are natural for scenario specifications
- If acknowledgment is not received within a reasonable amount of time, retransmit . . .

Can we extend the analysis techniques to the timed setting?

Unfortunately, most of the results are negative.

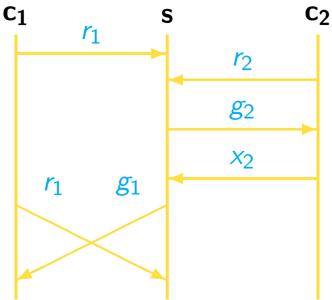
MSCs

Two clients and a server

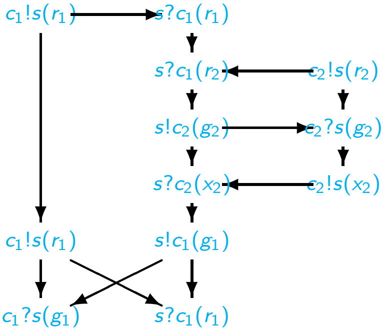


MSCs as Partial Orders

Two clients and a server,

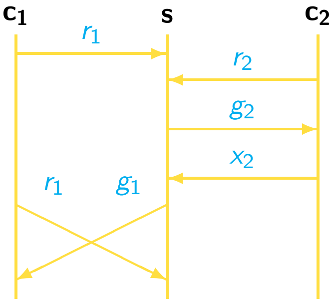


and a partial order representation

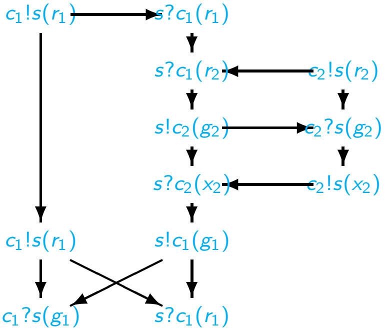


MSCs as Partial Orders

Two clients and a server,



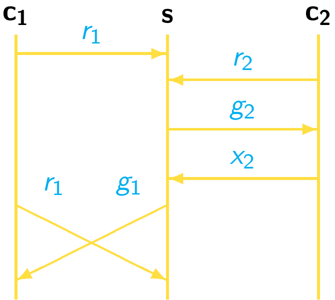
and a partial order representation



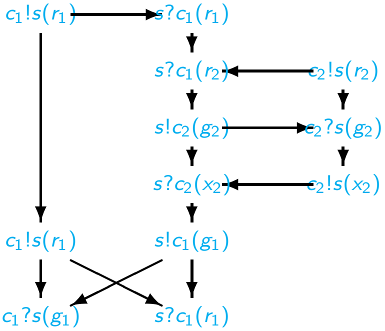
- All channels are assumed to be FIFO.

MSCs as Partial Orders

Two clients and a server,

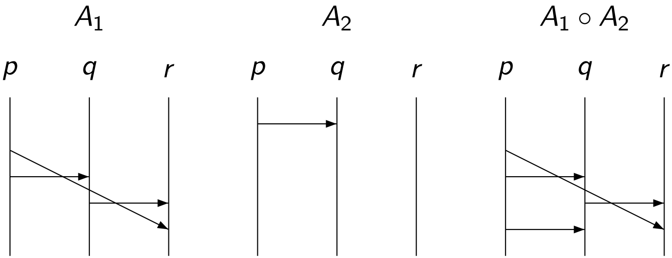


and a partial order representation

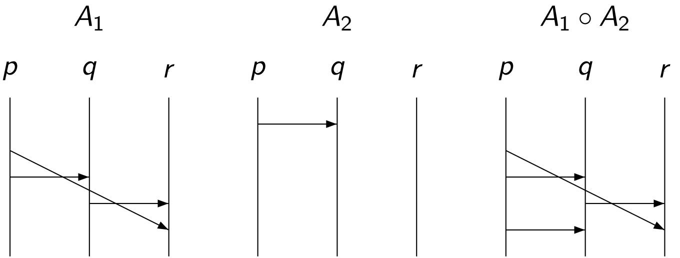


- All channels are assumed to be FIFO.
- An MSC can be regenerated from any one sequentialization.

Concatenation of MSCs



Concatenation of MSCs



p!r p!q q?p q!r r?q p!q q?p r?p

is a sequentialization of of $A_1 \circ A_2$.

Message Sequence Graphs

- A finite state automaton

Message Sequence Graphs

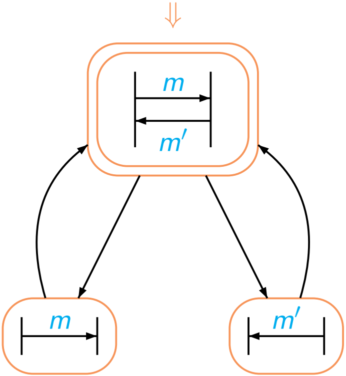
- A finite state automaton
- Each state is labelled by an MSC

Message Sequence Graphs

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

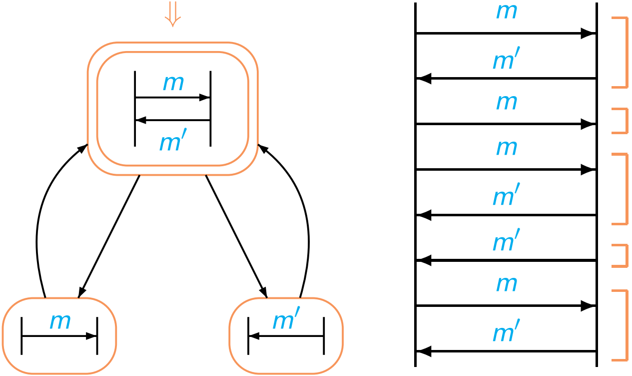
Message Sequence Graphs

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



Message Sequence Graphs

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

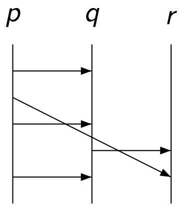


Boundedness

A sequentialization of an MSC is B-bounded if no channel has more than B messages at any point.

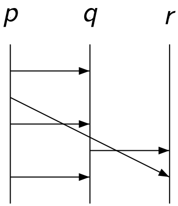
Boundedness

A sequentialization of an MSC is B-bounded if no channel has more than B messages at any point.



Boundedness

A sequentialization of an MSC is B-bounded if no channel has more than B messages at any point.



The linearization

p!q q?p p!r p!q q?p q!r r?q p!q q?p r?p

is 1-bounded while the linearization

p!q p!r p!q p!q q?p q?p p!r q?p r?q r?p

is 3-bounded.

Boundedness ...

An MSC is existentially B-bounded if one of its sequentializations is B-bounded.

Boundedness ...

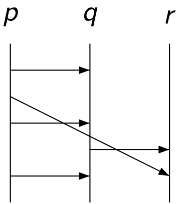
An MSC is existentially B-bounded if one of its sequentializations is B-bounded.

An MSC is universally B-bounded if all of its sequentializations are B-bounded.

Boundedness ...

An MSC is existentially B-bounded if one of its sequentializations is B-bounded.

An MSC is universally B-bounded if all of its sequentializations are B-bounded.



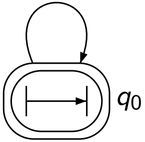
An existentially 1-bounded and universally 3-bounded MSC.

Boundedness ...

An MSG is existentially B-bounded if every MSC it generates is existentially B-bounded.

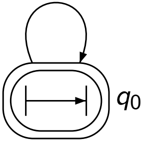
Boundedness ...

An MSG is existentially B-bounded if every MSC it generates is existentially B-bounded.



Boundedness ...

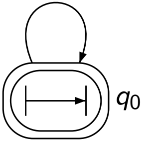
An MSG is existentially B-bounded if every MSC it generates is existentially B-bounded.



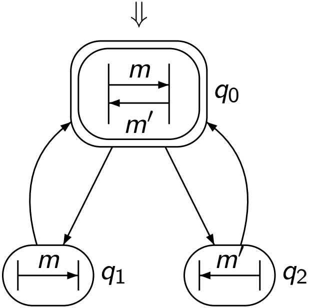
An MSG is universally B-bounded if every MSC it generates is universally B-bounded.

Boundedness ...

An MSG is existentially B-bounded if every MSC it generates is existentially B-bounded.

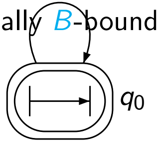


An MSG is universally B-bounded if every MSC it generates is universally B-bounded.

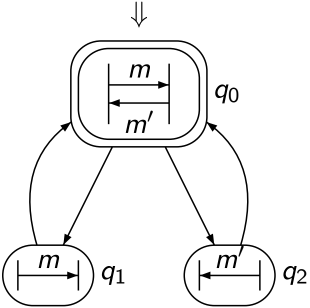


Boundedness ...

An MSG is existentially bounded if there exists a B such that every MSC it generates is existentially B-bounded.



An MSG is universally bounded if there exists a B such that every MSC it generates is B-bounded.



Deciding Boundedness

Deciding Boundedness

- Every MSG is existentially B-bounded for some B.

Deciding Boundedness

- Every MSG is existentially B-bounded for some B.
- Checking whether an MSG is existentially B-bounded for a given B is decidable.

Deciding Boundedness

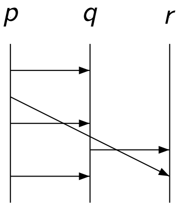
- Every MSG is existentially B-bounded for some B.
- Checking whether an MSG is existentially B-bounded for a given B is decidable.
- Checking whether an MSG is universally B-bounded for a given B is decidable.

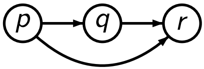
Deciding Boundedness

- Every MSG is existentially B-bounded for some B.
- Checking whether an MSG is existentially B-bounded for a given B is decidable.
- Checking whether an MSG is universally B-bounded for a given B is decidable.
- Checking whether an MSG is bounded is decidable.

Communication graph of an MSC

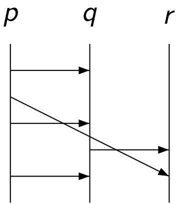
Nodes are the processes. An edge from p to q if there is a message from p to q.

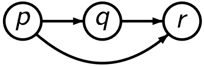




Communication graph of an MSC

Nodes are the processes. An edge from p to q if there is a message from p to q.





An MSG is bounded if and only if every the MSC generated by every loop has a communication graph that is a disjoint union of SCCs.

Adding time to scenarios

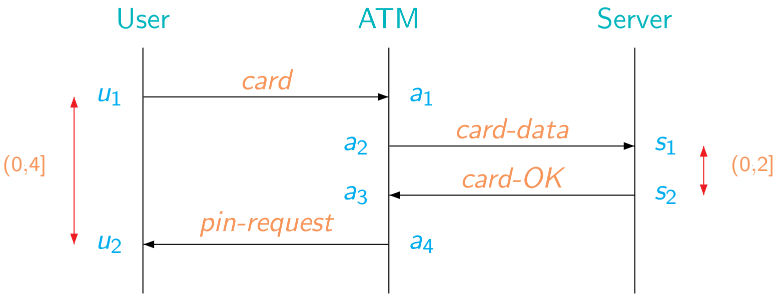
Adding time to scenarios

- Time constrained MSCs
  - MSCs with timing constraints between events

Adding time to scenarios

- Time constrained MSCs
  - MSCs with timing constraints between events
- Time constrained Message Sequence Graphs
  - Generate infinite families of time constrained MSCs

MSCs with time constraints



Time Constrained MSCs

- Associate time interval constraints with pairs of events

Time Constrained MSCs

- Associate time interval constraints with pairs of events
- If $(e, e') \longmapsto [l, u]$, then the time between occurrence of e and e' must be between l and u

Time Constrained MSCs

- Associate time interval constraints with pairs of events
- If $(e, e') \mapsto [l, u]$, then the time between occurrence of e and e' must be between l and u
- Intervals may be open, closed, half-open

Time Constrained MSCs

- Associate time interval constraints with pairs of events
- If $(e, e') \longmapsto [l, u]$, then the time between occurrence of e and e' must be between l and u
- Intervals may be open, closed, half-open
- Simplifying assumptions

Time Constrained MSCs

- Associate time interval constraints with pairs of events
- If $(e, e') \longmapsto [l, u]$, then the time between occurrence of e and e' must be between l and u
- Intervals may be open, closed, half-open
- Simplifying assumptions
  - Interval constraints are local to a process ...

Time Constrained MSCs

- Associate time interval constraints with pairs of events
- If $(e, e') \longmapsto [l, u]$, then the time between occurrence of e and e' must be between l and u
- Intervals may be open, closed, half-open
- Simplifying assumptions
  - Interval constraints are local to a process ...
    - Both e and e' lie on same process line

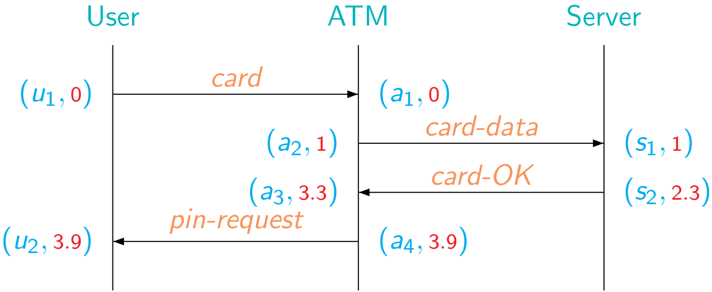
Time Constrained MSCs

- Associate time interval constraints with pairs of events
- If $(e, e') \mapsto [l, u]$, then the time between occurrence of e and e' must be between l and u
- Intervals may be open, closed, half-open
- Simplifying assumptions
  - Interval constraints are local to a process ...
    - Both e and e' lie on same process line
  - ... or across a single message

Time Constrained MSCs

- Associate time interval constraints with pairs of events
- If $(e, e') \mapsto [l, u]$, then the time between occurrence of e and e' must be between l and u
- Intervals may be open, closed, half-open
- Simplifying assumptions
  - Interval constraints are local to a process . . .
    - Both e and e' lie on same process line
  - . . . or across a single message
    - e is $p!q(m)$ and e' is corresponding receive $q?p(m)$

A timed behaviour



Timed MSCs

- Add timestamps to events on MSC, $\tau : E \rightarrow \mathbb{R}_{\geq 0}$

Timed MSCs

- Add timestamps to events on MSC, $\tau : E \rightarrow \mathbb{R}_{\geq 0}$
- All timestamps refer to same global time

Timed MSCs

- Add timestamps to events on MSC, $\tau : E \to \mathbb{R}_{\geq 0}$
- All timestamps refer to same global time
- Order of timestamps respects partial order on events

Timed MSCs

- Add timestamps to events on MSC, $\tau : E \to \mathbb{R}_{\geq 0}$
- All timestamps refer to same global time
- Order of timestamps respects partial order on events

- Linearizations of timed MSCs are timed words

Timed MSCs

- Add timestamps to events on MSC, $\tau : E \to \mathbb{R}_{\geq 0}$
- All timestamps refer to same global time
- Order of timestamps respects partial order on events

- Linearizations of timed MSCs are timed words
- Again, a single linearization suffices to reconstruct a timed MSC

Timed MSCs

- Add timestamps to events on MSC, $\tau : E \rightarrow \mathbb{R}_{\geq 0}$
- All timestamps refer to same global time
- Order of timestamps respects partial order on events

- Linearizations of timed MSCs are timed words
- Again, a single linearization suffices to reconstruct a timed MSC

- A timed MSC covers a TC-MSC if for each constraint $(e, e') \mapsto [l, u]$, $l \leq \tau(e') - \tau(e) \leq u$

Timed MSCs

- Add timestamps to events on MSC, $\tau : E \rightarrow \mathbb{R}_{\geq 0}$
- All timestamps refer to same global time
- Order of timestamps respects partial order on events

- Linearizations of timed MSCs are timed words
- Again, a single linearization suffices to reconstruct a timed MSC

- A timed MSC covers a TC-MSC if for each constraint $(e, e') \mapsto [l, u]$, $l \leq \tau(e') - \tau(e) \leq u$
  - Replace $\leq$ by $<$, as appropriate, for open, half-open intervals

Timed MSCs

- Add timestamps to events on MSC, $\tau : E \to \mathbb{R}_{\geq 0}$
- All timestamps refer to same global time
- Order of timestamps respects partial order on events

- Linearizations of timed MSCs are timed words
- Again, a single linearization suffices to reconstruct a timed MSC

- A timed MSC covers a TC-MSC if for each constraint $(e, e') \mapsto [l, u]$, $l \leq \tau(e') - \tau(e) \leq u$
  - Replace $\leq$ by $<$, as appropriate, for open, half-open intervals
- TC-MSC $T \Rightarrow L(T)$, set of timed MSCs that cover T

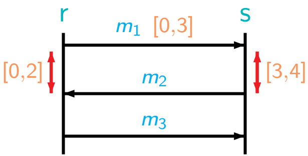
TC-MSCs and Timed MSCs

TC-MSCs and Timed MSCs

- The set of timed MSCs covering a TC-MSC may be empty.

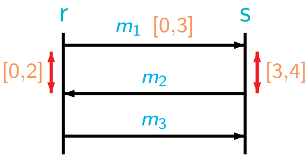
TC-MSCs and Timed MSCs

- The set of timed MSCs covering a TC-MSC may be empty.



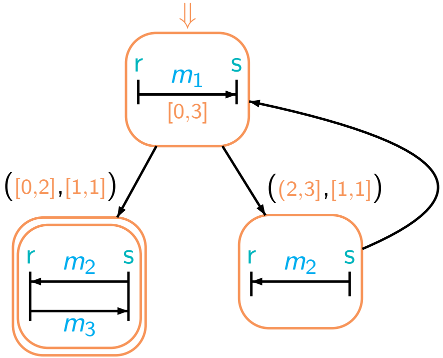
TC-MSCs and Timed MSCs

- The set of timed MSCs covering a TC-MSC may be empty.
- A TC-MSC is said to be realizable if it is covered by atleast one timed MSC.



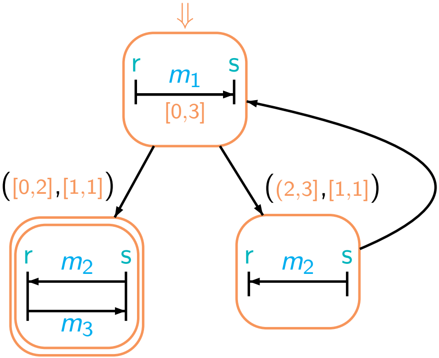
Time Constrained Message Sequence Graphs

- States labelled by time constrained MSCs
- Local constraints for each process along edges
- Legal paths in the automaton generate time constrained MSCs



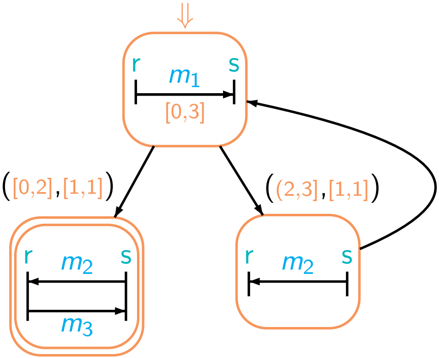
Time Constrained Message Sequence Graphs

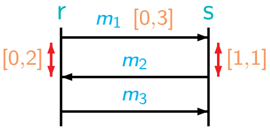
- States labelled by time constrained MSCs
- Local constraints for each process along edges
- Legal paths in the automaton generate time constrained MSCs



Time Constrained Message Sequence Graphs

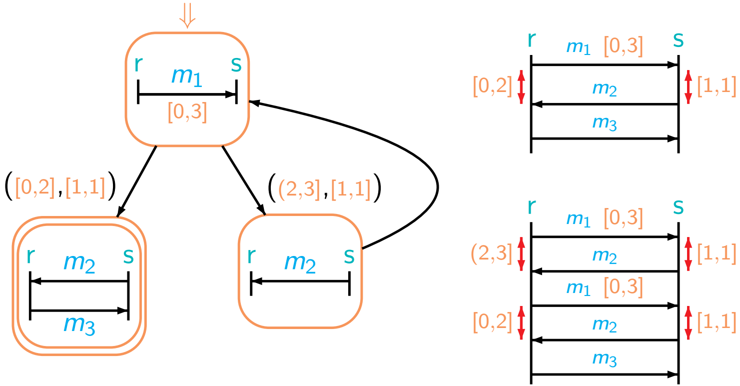
- States labelled by time constrained MSCs
- Local constraints for each process along edges
- Legal paths in the automaton generate time constrained MSCs





Time Constrained Message Sequence Graphs

- States labelled by time constrained MSCs
- Local constraints for each process along edges
- Legal paths in the automaton generate time constrained MSCs



Reachability

Reachability

Given a TC-MSG G and a state q in G, does there exist a path $q_0 q_1 \dots q_k = q$ from an initial state q_0 such that the TC-MSG generated by this path is realizable ?

Reachability

Given a TC-MSG G and a state q in G, does there exist a path $q_0 q_1 \dots q_k = q$ from an initial state q_0 such that the TC-MSG generated by this path is realizable ?

(The control state reachability problem for TC-MSGs.)

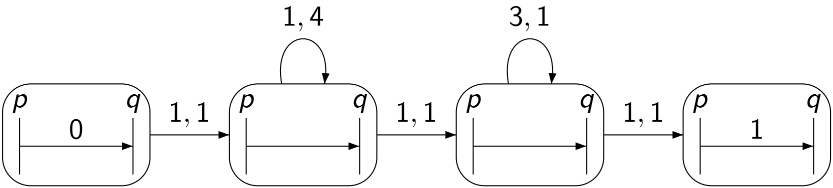
Reachability

Given a TC-MSG G and a state q in G, does there exist a path $q_0 q_1 \ldots q_k = q$ from an initial state q_0 such that the TC-MSG generated by this path is realizable ?

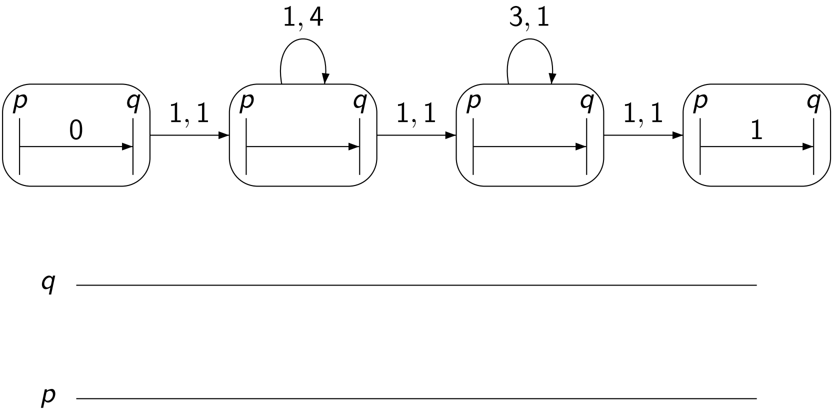
(The control state reachability problem for TC-MSGs.)

This problem is trivial for ordinary MSGs.

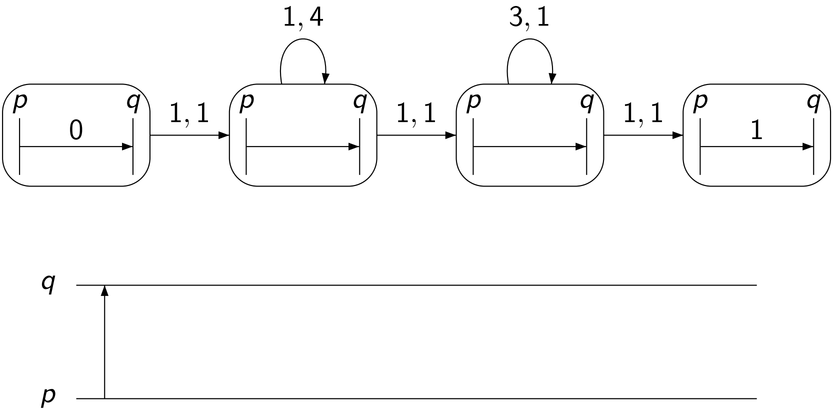
Reachability ...



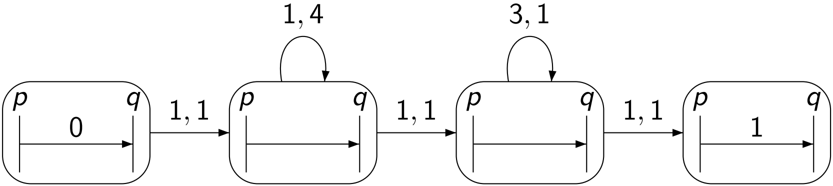
Reachability ...

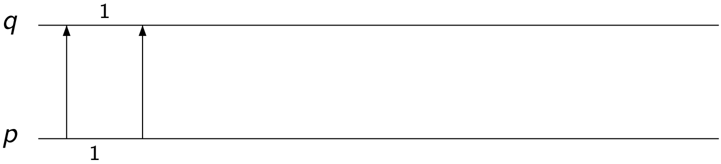


Reachability ...

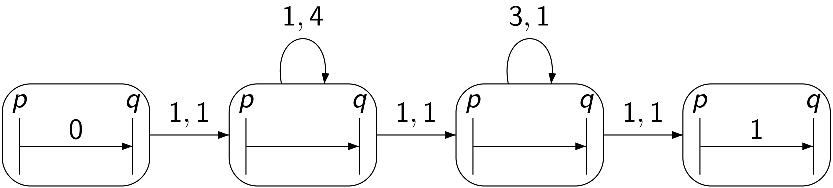


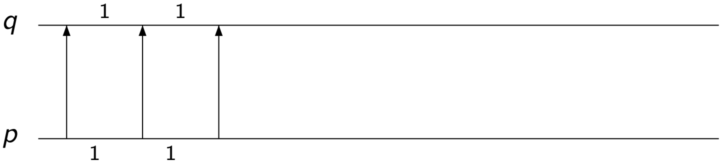
Reachability ...



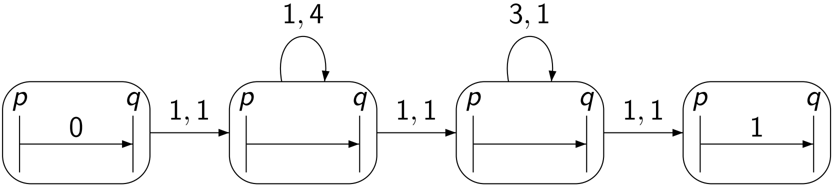


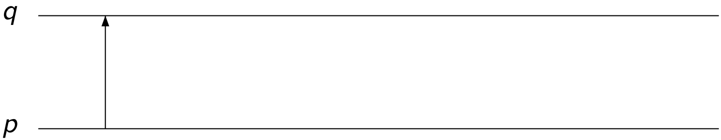
Reachability ...



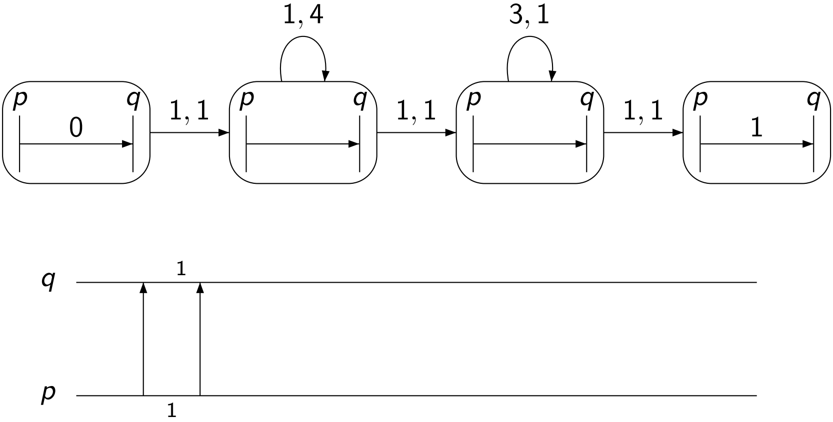


Reachability ...

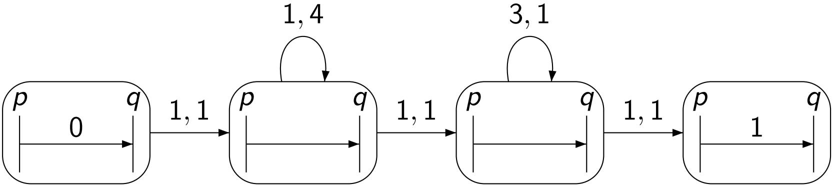


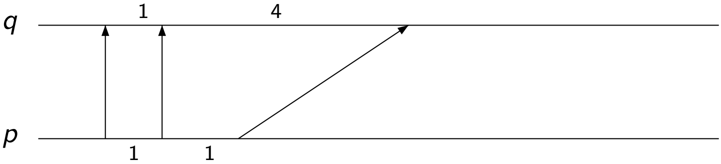


Reachability ...

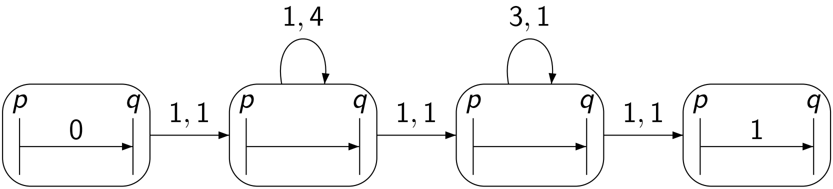


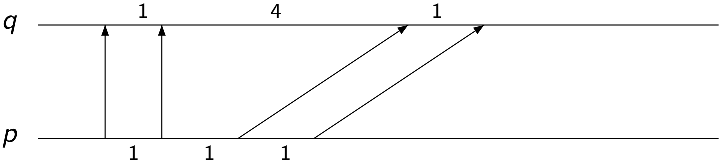
Reachability ...



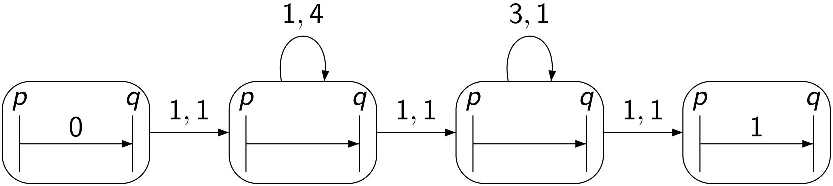


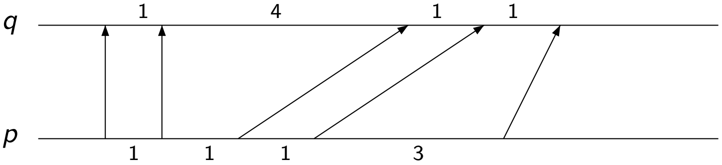
Reachability ...



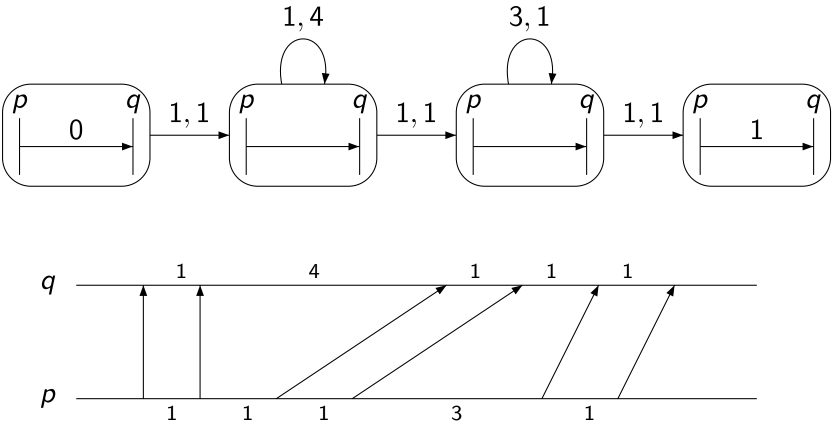


Reachability ...

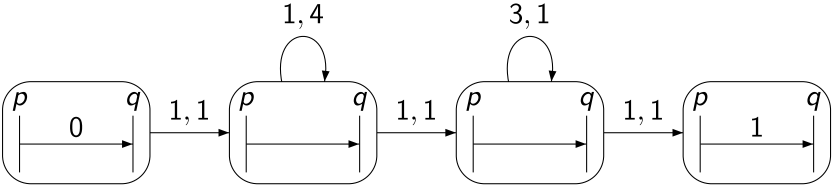


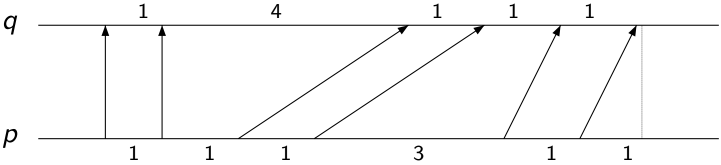


Reachability ...

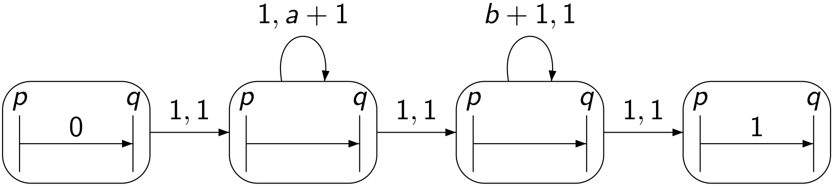


Reachability ...

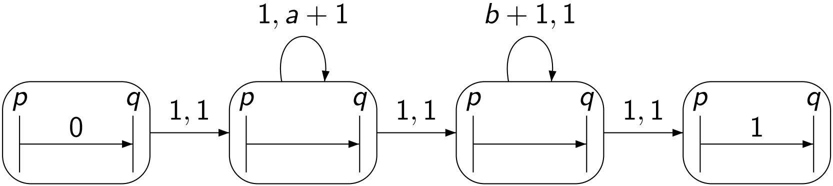




Reachability ...

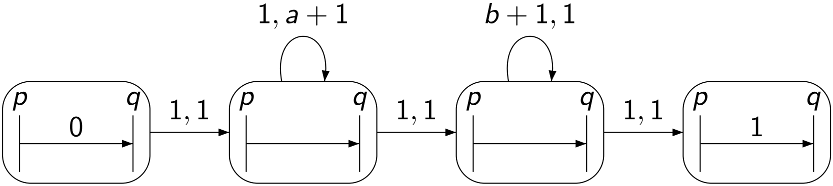


Reachability ...



- The first loop is to be executed k times and the second one l times such that $a.k - b.l = 1$.

Reachability ...



- The first loop is to be executed k times and the second one l times such that $a.k - b.l = 1$.
- Simple paths may not be realizable while those with loops may be.

Boundedness for Timed MSCs

- A timed MSC is universally B bounded if all its timed linearizations are B bounded.

Boundedness for Timed MSCs

- A timed MSC is universally B bounded if all its timed linearizations are B bounded.
- A timed MSC is existentially B bounded if it has at least one timed linearization that is B bounded.

Boundedness for Timed MSCs

- A timed MSC is universally B bounded if all its timed linearizations are B bounded.
- A timed MSC is existentially B bounded if it has at least one timed linearization that is B bounded.
- A TC-MSC is (universally/existentially) B bounded if all its timed realizations are (universally/existentially) B bounded.

Boundedness for Timed MSCs

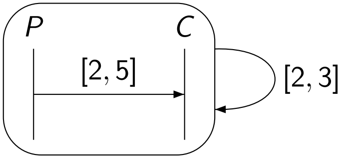
- A timed MSC is universally B bounded if all its timed linearizations are B bounded.
- A timed MSC is existentially B bounded if it has at least one timed linearization that is B bounded.
- A TC-MSC is (universally/existentially) B bounded if all its timed realizations are (universally/existentially) B bounded.
- A TC-MSG is (universally/existentially) bounded if there is a B such that all the TC-MSCs realizing it are (universally/existentially) B bounded.

Boundedness ...

Time constraints may ensure boundedness.

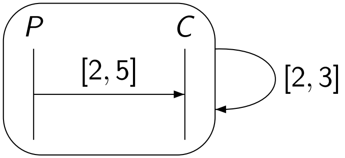
Boundedness ...

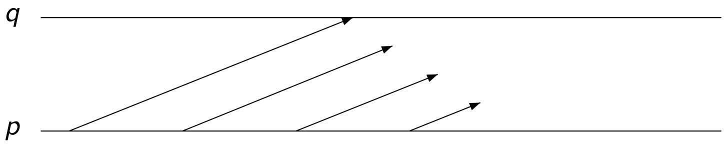
Time constraints may ensure boundedness.



Boundedness ...

Time constraints may ensure boundedness.



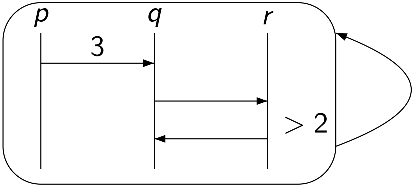


Boundedness ...

Time constraints may ensure boundedness.

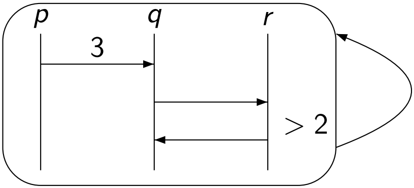
Boundedness ...

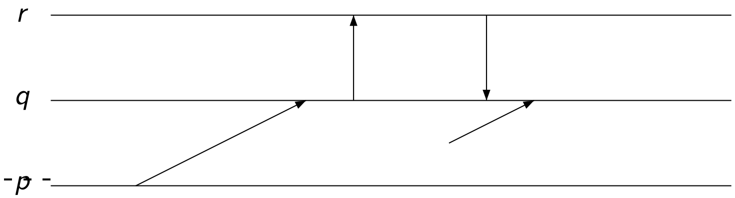
Time constraints may ensure boundedness.



Boundedness ...

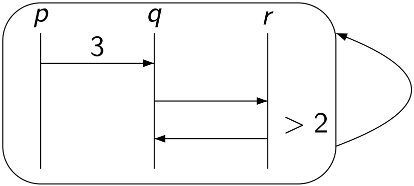
Time constraints may ensure boundedness.

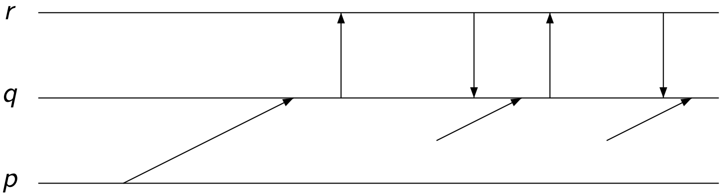




Boundedness ...

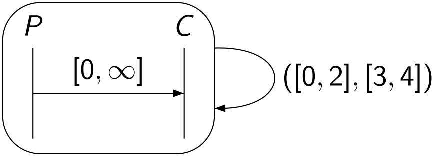
Time constraints may ensure boundedness.





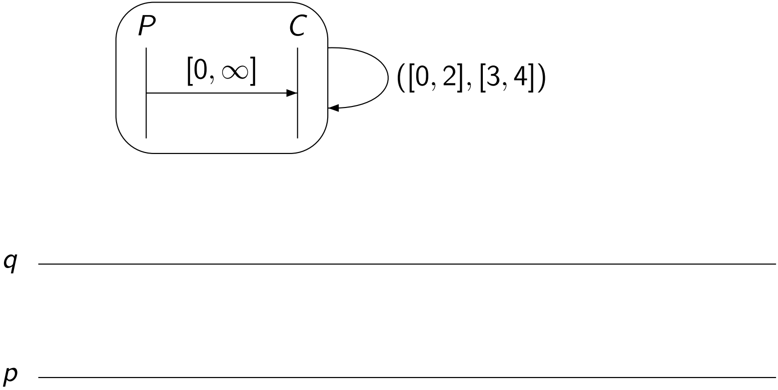
Boundedness ...

Time contraints may rule out existential boundedness.



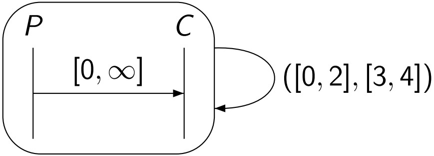
Boundedness ...

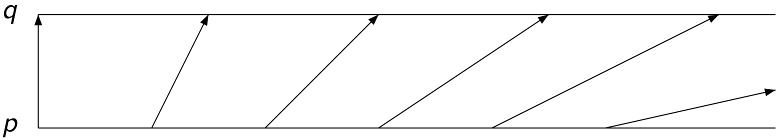
Time contraints may rule out existential boundedness.



Boundedness ...

Time contraints may rule out existential boundedness.





Undecidability of Reachability

We show that 2 counter machines can be simulated using TC-MSGs.

Undecidability of Reachability

We show that 2 counter machines can be simulated using TC-MSGs.

- Each *instruction* is coded by a node in the TC-MSG.

Undecidability of Reachability

We show that 2 counter machines can be simulated using TC-MSGs.

- Each *instruction* is coded by a node in the TC-MSG.
- Each counter c is maintained using 2 processes p_c and q_c.

Undecidability of Reachability

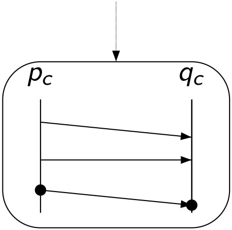
We show that 2 counter machines can be simulated using TC-MSGs.

- Each *instruction* is coded by a node in the TC-MSG.
- Each counter c is maintained using 2 processes p_c and q_c.
- In any run ending at a particular node, the difference between the time-stamp on the last q_c event and the last p_c event records the value of c.

Undecidability of Reachability

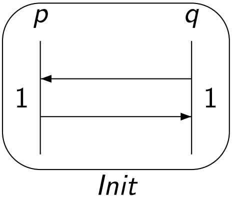
We show that 2 counter machines can be simulated using TC-MSGs.

- Each *instruction* is coded by a node in the TC-MSG.
- Each counter c is maintained using 2 processes p_c and q_c.
- In any run ending at a particular node, the difference between the time-stamp on the last q_c event and the last p_c event records the value of c.



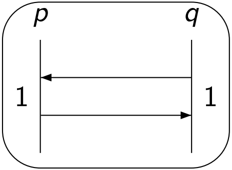
The reduction

- Initialization of the counter value to 0



The reduction

- Initialization of the counter value to 0

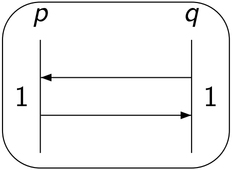


Init

- Keep counter values as it is (Freeze).

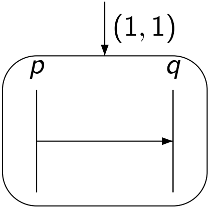
The reduction

- Initialization of the counter value to 0



Init

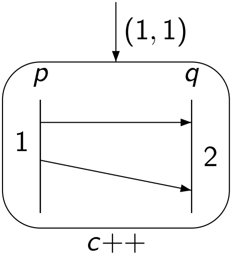
- Keep counter values as it is (Freeze).



Freeze

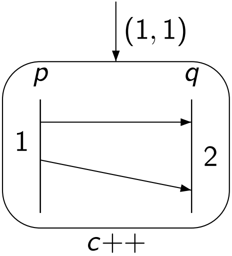
The Reduction ...

- Increment the counter c.



The Reduction ...

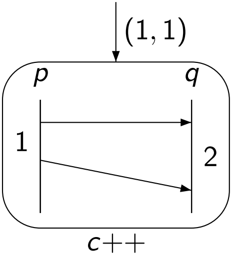
- Increment the counter c.



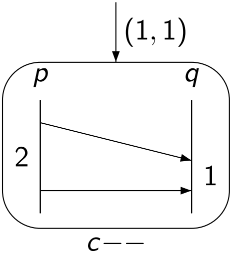
- Decrement the counter c.

The Reduction ...

- Increment the counter c.



- Decrement the counter c.

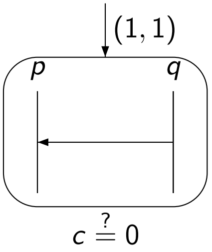


The Reduction ...

- Check if the counter is 0.

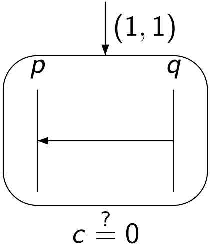
The Reduction ...

- Check if the counter is 0.



The Reduction ...

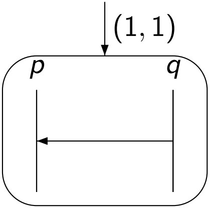
- Check if the counter is 0.



- Check if the counter is greater than 0.

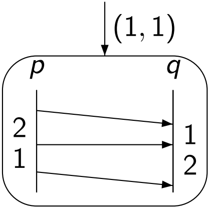
The Reduction ...

- Check if the counter is 0.



$c \stackrel{?}{=} 0$

- Check if the counter is greater than 0.



$c \stackrel{?}{>} 0$

The Reduction ...

- The counter machine is assumed to be deterministic.

The Reduction ...

- The counter machine is assumed to be deterministic.
- It either has a finite run ending at the accept state or an infinite run.

The Reduction ...

- The counter machine is assumed to be deterministic.
- It either has a finite run ending at the accept state or an infinite run.
- The control state corresponding to the final state is reachable if and only if the counter machine halts.

The Reduction ...

- The counter machine is assumed to be deterministic.
- It either has a finite run ending at the accept state or an infinite run.
- The control state corresponding to the final state is reachable if and only if the counter machine halts.

The control state reachability problem for TC-MSGs is undecidable. The problem is undecidable even when there are no timing constraints on messages.

The Reduction ...

- The counter machine is assumed to be deterministic.
- It either has a finite run ending at the accept state or an infinite run.
- The control state corresponding to the final state is reachable if and only if the counter machine halts.

The control state reachability problem for TC-MSGs is undecidable. The problem is undecidable even when there are no timing constraints on messages.

The (language) emptiness problem for TC-MSGs is undecidable.

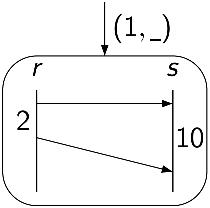
Reachability to Boundedess

Reachability to Boundedess

- Add two new processes r and s.

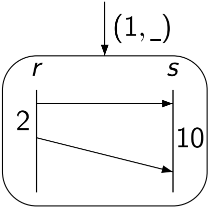
Reachability to Boundedess

- Add two new processes r and s.
- Augment the TC-MSC labelling each node with the following two messages



Reachability to Boundedess

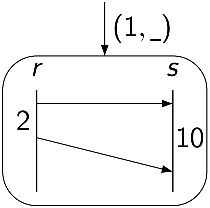
- Add two new processes r and s.
- Augment the TC-MSC labelling each node with the following two messages



- Label all the nonhalting states as accepting.

Reachability to Boundedess

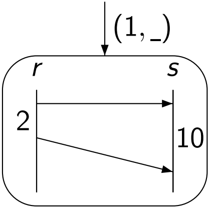
- Add two new processes *r* and *s*.
- Augment the TC-MSC labelling each node with the following two messages



- Label all the nonhalting states as accepting.
- If the counter machine halts then the language is finite and hence bounded.

Reachability to Boundedess

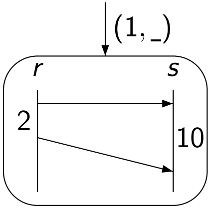
- Add two new processes r and s.
- Augment the TC-MSC labelling each node with the following two messages



- Label all the nonhalting states as accepting.
- If the counter machine halts then the language is finite and hence bounded.
- If the counter machine does not halt then the language is not even existentially bounded.

Reachability to Boundedess

- Add two new processes r and s.
- Augment the TC-MSC labelling each node with the following two messages



- Label all the nonhalting states as accepting.
- If the counter machine halts then the language is finite and hence bounded.
- If the counter machine does not halt then the language is not even existentially bounded.

Checking boundedness for TC-MSGs is undecidable

More Undecidability – 1

Are point intervals necessary to obtain undecidability?

More Undecidability – 1

Are point intervals necessary to obtain undecidability?

Reachability and Boundedness are undecidable even when all interval constraints are restricted to be open intervals.

More Undecidability – 1

Are point intervals necessary to obtain undecidability?

Reachability and Boundedness are undecidable even when all interval constraints are restricted to be open intervals.

- Use four processes p_l, q_l, p_u and q_u for each counter.

More Undecidability – 1

Are point intervals necessary to obtain undecidability?

Reachability and Boundedness are undecidable even when all interval constraints are restricted to be open intervals.

- Use four processes p_l, q_l, p_u and q_u for each counter.
- One pair maintains a lower bound on the value of the counter while the other maintains an upper bound.

More Undecidability – 1

Are point intervals necessary to obtain undecidability?

Reachability and Boundedness are undecidable even when all interval constraints are restricted to be open intervals.

- Use four processes p_l, q_l, p_u and q_u for each counter.
- One pair maintains a lower bound on the value of the counter while the other maintains an upper bound.
- The value $p_l - q_l$ is used to ensure that the $C--$ operation is *permissible* only if the counter is nonzero.

More Undecidability – 1

Are point intervals necessary to obtain undecidability?

Reachability and Boundedness are undecidable even when all interval constraints are restricted to be open intervals.

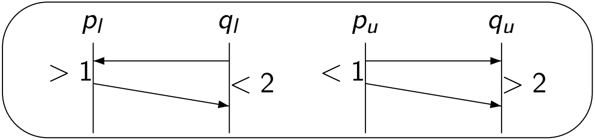
- Use four processes p_l, q_l, p_u and q_u for each counter.
- One pair maintains a lower bound on the value of the counter while the other maintains an upper bound.
- The value $p_l - q_l$ is used to ensure that the $C--$ operation is *permissible* only if the counter is nonzero.
- The value of $p_u - q_u$ is used to check for 0.

Open Intervals ...

Initialize the counter to 0.

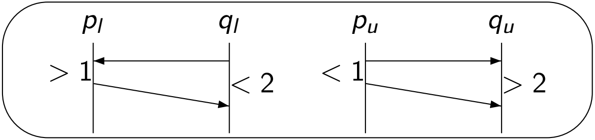
Open Intervals ...

Initialize the counter to 0.

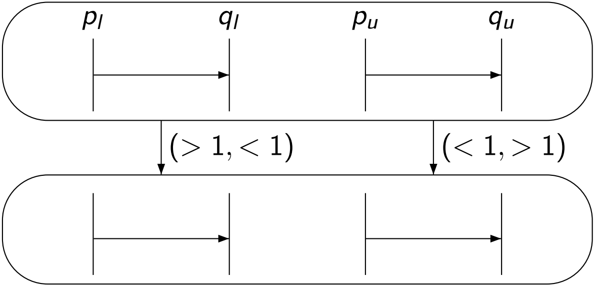


Open Intervals ...

Initialize the counter to 0.



Composition between Nodes

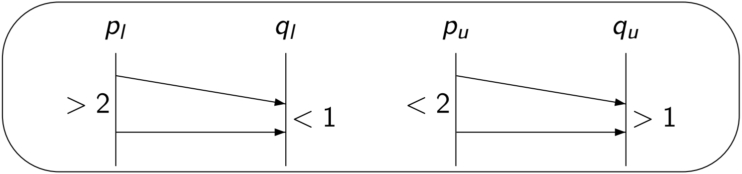


Open intervals ...

The decrement instruction

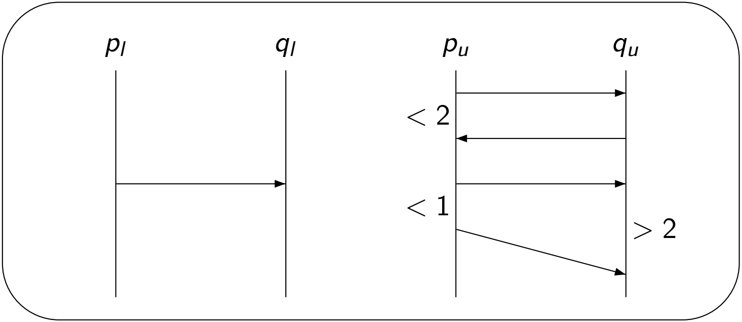
Open intervals ...

The decrement instruction



Open interval ...

Check for 0



More Undecidability – 2

What about the reachability problem for channel bounded TC-MSGs?

More Undecidability – 2

What about the reachability problem for channel bounded TC-MSGs?

The reachability problem for channel bounded TC-MSGs is also undecidable.

Single process as a time keeper

- Two processes are used to simulate a counter.

Single process as a time keeper

- Two processes are used to simulate a counter.
- Restrict constraints across nodes to only one fixed process (across all transitions).

Single process as a time keeper

- Two processes are used to simulate a counter.
- Restrict constraints across nodes to only one fixed process (across all transitions).
- A reasonable restriction.

Single process as a time keeper

- Two processes are used to simulate a counter.
- Restrict constraints across nodes to only one fixed process (across all transitions).
- A reasonable restriction.

  A controller or scheduler process that dictates timing across different phases of the protocol.

Single process as a time keeper

- Two processes are used to simulate a counter.
- Restrict constraints across nodes to only one fixed process (across all transitions).
- A reasonable restriction.

  A controller or scheduler process that dictates timing across different phases of the protocol.

Even with the restriction that constraints across nodes are permitted only on a fixed process, the reachability and boundedness problems for TC-MSGs remain undecidable.

More Undecidability – 3

Let p be the time-keeper. We use two processes q^- and q^+.

More Undecidability – 3

Let p be the time-keeper. We use two processes q^- and q^+.

1. The time difference between the last events in p and q^- is a lower bound on the value of the clock.

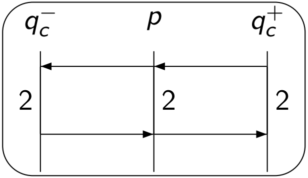
More Undecidability – 3

Let p be the time-keeper. We use two processes q^- and q^+.

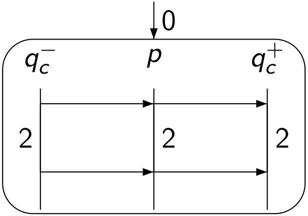
1. The time difference between the last events in p and q^- is a lower bound on the value of the clock.
2. The time difference between the last events in q^+ and p is an upper bound on the value of the clock.

More Undecidability - 3

Initialize

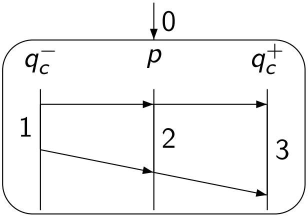


Freeze

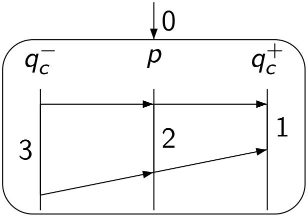


More Undecidability – 3

Increment

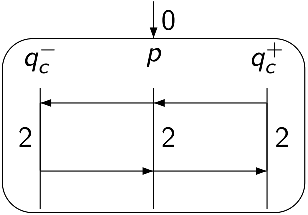


Decrement



More Undecidability – 3

Check for Zero



Locally synchronized MSGs

- Construct communication graph for an MSC
  One node per process, edge $p \longrightarrow q$ iff p sends a message to q

Locally synchronized MSGs

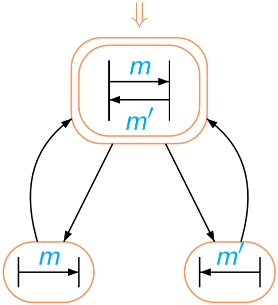
- Construct communication graph for an MSC
  One node per process, edge $p \longrightarrow q$ iff p sends a message to q
- For each loop, communication graph is one strongly connected component plus isolated vertices

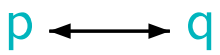
Locally synchronized MSGs

- Construct communication graph for an MSC
  One node per process, edge $p \longrightarrow q$ iff p sends a message to q
- For each loop, communication graph is one strongly connected component plus isolated vertices
- In each loop, every message is "acknowledged"

Locally synchronized MSGs

- Construct communication graph for an MSC
  One node per process, edge $p \longrightarrow q$ iff p sends a message to q
- For each loop, communication graph is one strongly connected component plus isolated vertices
- In each loop, every message is "acknowledged"





Locally Synchronized TC-MSGs

Every locally synchronized MSG generates a universally bounded language.

Locally Synchronized TC-MSGs

Every locally synchronized MSG generates a universally bounded language.

- For locally synchronized TC-MSGs the boundedness problem is trivially decidable.

Locally Synchronized TC-MSGs

Every locally synchronized MSG generates a universally bounded language.

- For locally synchronized TC-MSGs the boundedness problem is trivially decidable.
- The reachability problem for locally synchronized TC-MSGs is decidable.

Locally Synchronized TC-MSGs

Every locally synchronized MSG generates a universally bounded language.

- For locally synchronized TC-MSGs the boundedness problem is trivially decidable.
- The reachability problem for locally synchronized TC-MSGs is decidable.
  - The untimed behaviour is regular.

Locally Synchronized TC-MSGs

Every locally synchronized MSG generates a universally bounded language.

- For locally synchronized TC-MSGs the boundedness problem is trivially decidable.
- The reachability problem for locally synchronized TC-MSGs is decidable.
  - The untimed behaviour is regular.
  - The number of active clocks is bounded.

Locally Synchronized TC-MSGs

Every locally synchronized MSG generates a universally bounded language.

- For locally synchronized TC-MSGs the boundedness problem is trivially decidable.
- The reachability problem for locally synchronized TC-MSGs is decidable.
  - The untimed behaviour is regular.
  - The number of active clocks is bounded.
  - Associate clocks to event labels. A clock can be reset everytime an associated event takes place.

Locally Synchronized TC-MSGs

Every locally synchronized MSG generates a universally bounded language.

- For locally synchronized TC-MSGs the boundedness problem is trivially decidable.
- The reachability problem for locally synchronized TC-MSGs is decidable.
  - The untimed behaviour is regular.
  - The number of active clocks is bounded.
  - Associate clocks to event labels. A clock can be reset everytime an associated event takes place.
  - Works like an event-clock automaton (upto some extra labelling).

Conclusions and future directions

- Analyzing timed constrained MSGs is difficult.

Conclusions and future directions

- Analyzing timed constrained MSGs is difficult.
- The culprit seems to be the use of a global time in the semantics.

Conclusions and future directions

- Analyzing timed constrained MSGs is difficult.
- The culprit seems to be the use of a global time in the semantics.
- Consider local time/clock drift/...

Conclusions and future directions

- Analyzing timed constrained MSGs is difficult.
- The culprit seems to be the use of a global time in the semantics.
- Consider local time/clock drift/...
- Easier to formulate CFMs with local time.

Conclusions and future directions

- Analyzing timed constrained MSGs is difficult.
- The culprit seems to be the use of a global time in the semantics.
- Consider local time/clock drift/...
- Easier to formulate CFMs with local time.
- ... many undecidability results even with local time.

Conclusions and future directions

- Analyzing timed constrained MSGs is difficult.
- The culprit seems to be the use of a global time in the semantics.
- Consider local time/clock drift/...
- Easier to formulate CFMs with local time.
- ... many undecidability results even with local time.

Conclusions and future directions

- Analyzing timed constrained MSGs is difficult.
- The culprit seems to be the use of a global time in the semantics.
- Consider local time/clock drift/...
- Easier to formulate CFMs with local time.
- ... many undecidability results even with local time.

Thank you.

Edge Constraint free TC-MSGs

Consider TC-MSGs where there are no time constraints associated with transitions between nodes.

Edge Constraint free TC-MSGs

Consider TC-MSGs where there are no time constraints associated with transitions between nodes.

- The control state reachability problem is decidable. A path is realizable if and only if each node in the path is realizable.

Edge Constraint free TC-MSGs

Consider TC-MSGs where there are no time constraints associated with transitions between nodes.

- The control state reachability problem is decidable. A path is realizable if and only if each node in the path is realizable.
- The boundedness problem is still open. Time constraints can enforce boundedness.