Decidability of Timed Communicating Automata

L. Clemente, University of Warsaw

Praha, July 2018

Summary

1. The model: Timed communicating automata (TCA).
2. The problem: control-state reachability.
3. Solution technique: quantifier elimination, cyclic order atoms.

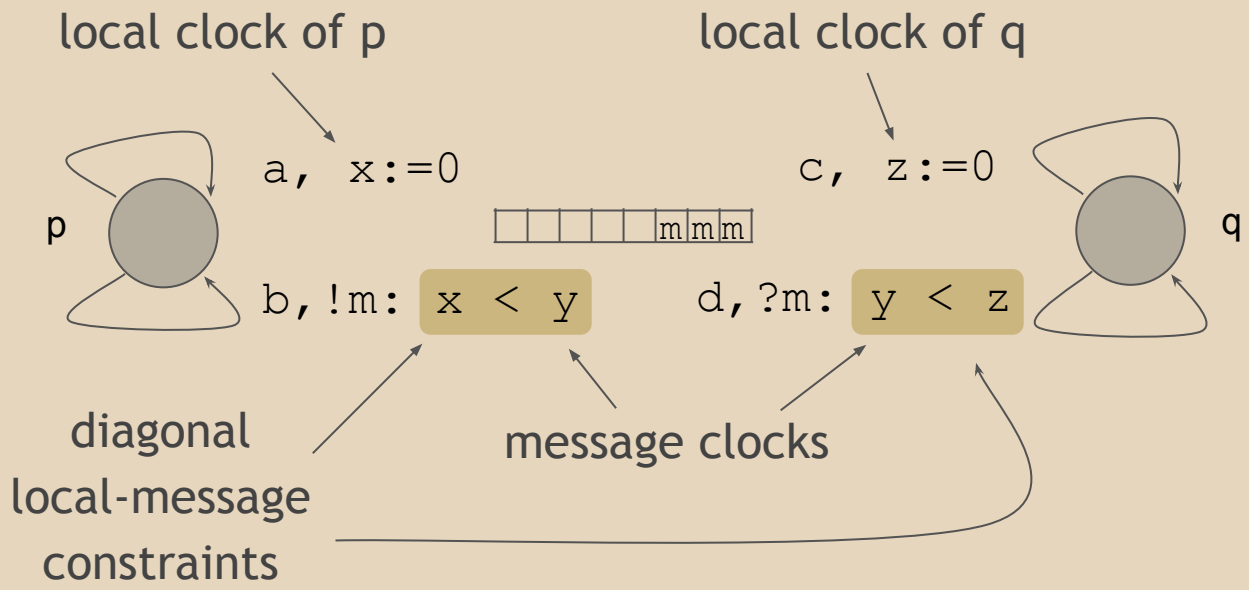
Timed communicating automata (TCA)

Networks of timed automata communicating by the asynchronous exchange of messages over FIFO queues.

- The time domain is *dense*.
- Each timed automaton controls its set of *local* clocks.
- Messages are equipped with dense *message* clocks ***NEW***.
- Diagonal constraints: local-local, *local-message* ***NEW***, message-message.
- All clocks evolve at the same rate.

Control state reachability: Given a network of TCA, and for each automaton its initial and final state, decide whether there is a run starting and ending with empty channels.

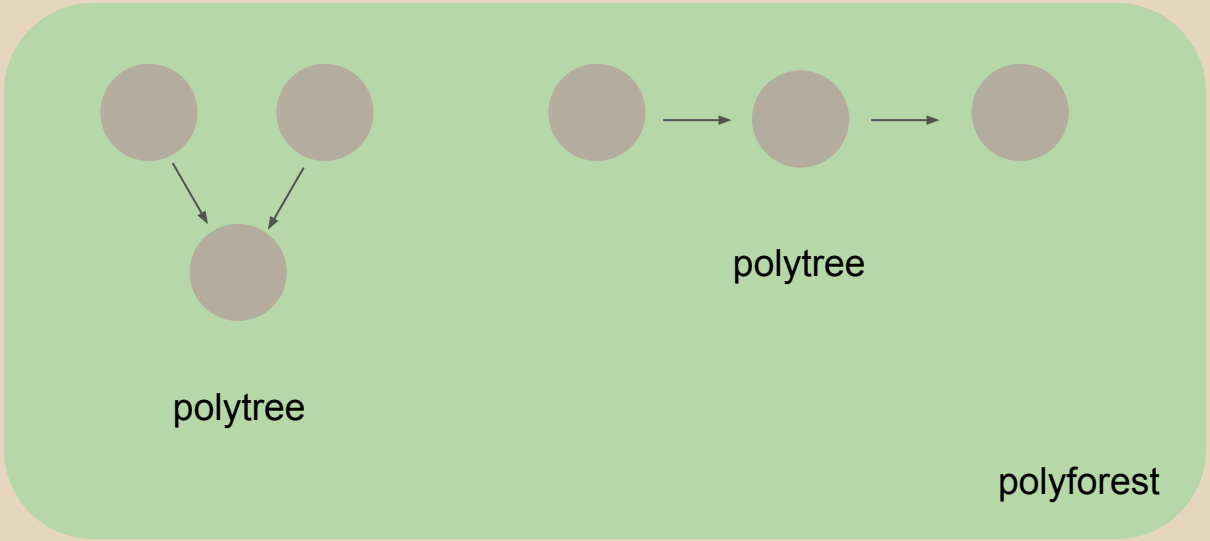
TCA example

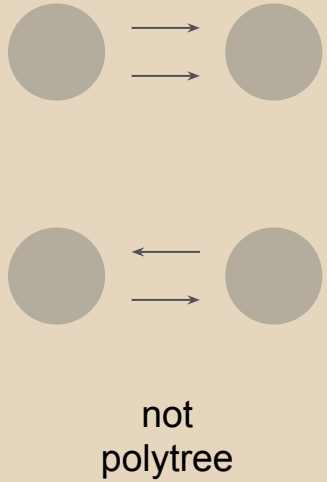

local clock of p
local clock of q
a, x:=0
c, z:=0
p
q
m m m
b,!m: x < y
d,?m: y < z
diagonal
local-message
constraints
message clocks

Communication topology

Polytree: no undirected cycles.

Polyforest: disjoint union of polytrees.





Main result

Characterisation of communication topologies with decidable reachability.

Theorem. Reachability is decidable iff the communication topology is a polyforest and for each polytree therein there is at most one channel with integer inequality tests.

Undecidability follows from [C, Herbreteau, Stainer, Sutre'13].
In the following, we focus on decidability for *timed channels*.

Related works

- Communicating automata (untimed) [Pachl'82; Brand, Zafiropulo'83].
  - Decidable for polyforest topologies.
- Communicating timed automata [Krčal, Yi'06].
  - Undecidable with two *urgent* channels, decidable with one.
- Communicating timed processes [C, Herbreteau, Stainer, Sutre'13].
  - Decidable for polyforest topologies with at most one urgent channel per comp.
- Timed lossy channel systems [Abdulla, Atig, Cederberg'12].
  - Non-diagonal constraints. Decidable.
- Communicating timed processes [Abdulla, Atig, Krishna'17].
  - Non-diagonal constraints. *Discrete time*.
  - Undecidable with two timed channels (with inequality constraints).
  - Decidable with one timed channel.
  - Undecidable with global clocks.

Decidability of TCA

1. ***NEW*** Reduce to the more constrained *simple TCA*:
    a. The initial value of message clock(s) is `0`.
    b. Reception constraints are either
        i. Integral non-diagonal: `x ~ k`, or
        ii. Fractional equality: `{y} = {z}`.

    Achieved via the method of *quantifier elimination*.
2. Desynchronised semantics (receivers ahead of senders) [Pachl'82].
3. Rendezvous semantics (handshaking communication → no channels) [ib.].
4. ***NEW*** Simulate 2,3 with register automata with counters (RAC).
    a. Counters keep track of the integral desynchronisation.
    b. Registers keep track of fractional values with *cyclic order atoms*.

Reduction to *simple* TCA

S*imple TCA*: The initial value of message clock(s) is `0`.
Reception constraints are either

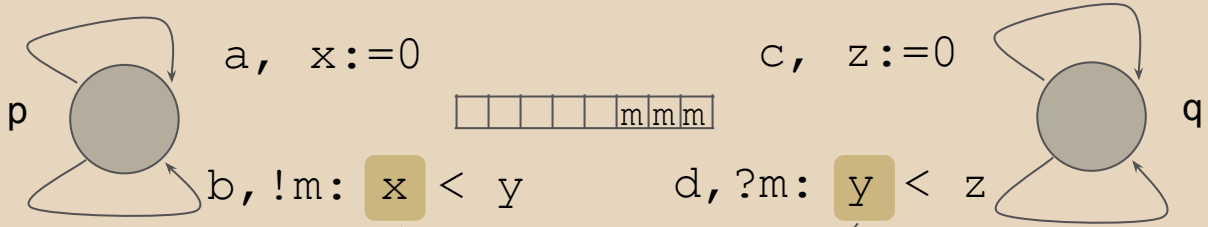
- Integral non-diagonal: `x ~ k`, or
- Fractional equality: `{y} = {z}`.

This is achieved in a number of steps.

1. Restricting transmission to *copy-send* (send copies of local clocks).
    - Quantifier elimination.
2. Send and receive constraints are atomic (i.e., only one conjunct).
3. Send `y = 0` and receive `x = y`.
4. Send `y = 0` and receive `y ~ k` (classical) and `{x} = {y}` (fractional).

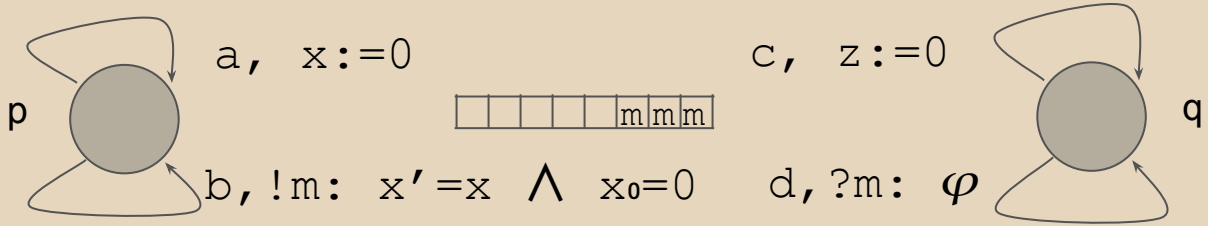
Quantifier elimination for TCA

Objective: The sender always sends copies of local clocks.



Local clocks: x, z.
Message clocks: y.

$$\varphi \equiv \exists y.\ x' - x_0 < y \wedge y + x_0 < z \Leftrightarrow \varphi' \equiv x' < z$$



Local clocks: x, z. Message clocks:

- x' (copy of x)
- x_0 (zero upon send).

Quantifier elimination for TCA

More generally: `!m:ψp` and `?m:ψq`. Before:

$$\psi^{\mathtt{p}} \equiv \bigwedge_{(i,j)\in I^{\mathtt{p}}} \mathrm{x}_i^{\mathtt{pq}} - \mathrm{x}_j^{\mathtt{p}} \sim_{ij}^{\mathtt{p}} k_{ij}^{\mathtt{p}} \;\wedge \bigwedge_{(i,j)\in I^{\mathtt{pq}}} \mathrm{x}_i^{\mathtt{pq}} - \mathrm{x}_j^{\mathtt{pq}} \sim_{ij}^{\mathtt{pq}} k_{ij}^{\mathtt{pq}} \;\wedge \qquad \text{(inequality)}$$

$$\bigwedge_{(i,j)\in J^{\mathtt{p}}} \mathrm{x}_i^{\mathtt{pq}} - \mathrm{x}_j^{\mathtt{p}} \equiv_M h_{ij}^{\mathtt{p}} \;\wedge \bigwedge_{(i,j)\in J^{\mathtt{pq}}} \mathrm{x}_i^{\mathtt{pq}} - \mathrm{x}_j^{\mathtt{pq}} \equiv_M h_{ij}^{\mathtt{pq}} \;\wedge \qquad \text{(modular)}$$

$$\bigwedge_{(i,j)\in K^{\mathtt{p}}} \mathrm{y}_i^{\mathtt{pq}} \approx_{ij}^{\mathtt{p}} \mathrm{y}_j^{\mathtt{p}} \;\wedge \bigwedge_{(i,j)\in K^{\mathtt{pq}}} \mathrm{y}_i^{\mathtt{pq}} \approx_{ij}^{\mathtt{pq}} \mathrm{y}_j^{\mathtt{pq}}, \text{ and} \qquad \text{(order)}$$

$$\psi^{\mathtt{q}} \equiv \bigwedge_{(i,j)\in I^{\mathtt{q}}} \mathrm{x}_i^{\mathtt{pq}} - \mathrm{x}_j^{\mathtt{q}} \sim_{ij}^{\mathtt{q}} k_{ij}^{\mathtt{q}} \;\wedge \bigwedge_{(i,j)\in J^{\mathtt{q}}} \mathrm{x}_i^{\mathtt{pq}} - \mathrm{x}_j^{\mathtt{q}} \equiv_M h_{ij}^{\mathtt{q}} \;\wedge \bigwedge_{(i,j)\in K^{\mathtt{q}}} \mathrm{y}_i^{\mathtt{pq}} \approx_{ij}^{\mathtt{q}} \mathrm{y}_j^{\mathtt{q}},$$

Quantifier elimination for TCA

More generally: `!m:ψp` and `?m:ψq`. After:

$$\psi_0^{\mathtt{q}} \equiv (\exists \bar{\mathtt{x}}^{\mathtt{pq}} \cdot \psi^{\mathtt{q}}_{\bar{\mathtt{x}}^{\mathtt{pq}}}) \wedge (\exists \bar{\mathtt{y}}^{\mathtt{pq}} \cdot \psi^{\mathtt{q}}_{\bar{\mathtt{y}}^{\mathtt{pq}}}), \text{ where}$$

$$\begin{aligned}
\psi^{\mathtt{q}}_{\bar{\mathtt{x}}^{\mathtt{pq}}} \equiv{}& \bigwedge_{(i,j)\in I^{\mathtt{p}}} \mathtt{x}_i^{\mathtt{pq}} - (\hat{\mathtt{x}}_j^{\mathtt{pq}} - \hat{\mathtt{x}}_0^{\mathtt{pq}}) \sim^{\mathtt{p}}_{ij} k^{\mathtt{p}}_{ij} \wedge \bigwedge_{(i,j)\in I^{\mathtt{pq}}} \mathtt{x}_i^{\mathtt{pq}} - \mathtt{x}_j^{\mathtt{pq}} \sim^{\mathtt{pq}}_{ij} k^{\mathtt{pq}}_{ij} \wedge \bigwedge_{(i,j)\in I^{\mathtt{q}}} (\mathtt{x}_i^{\mathtt{pq}} + \hat{\mathtt{x}}_0^{\mathtt{pq}}) - \mathtt{x}_j^{\mathtt{q}} \sim^{\mathtt{q}}_{ij} k^{\mathtt{q}}_{ij} \wedge \\
& \bigwedge_{(i,j)\in J^{\mathtt{p}}} \mathtt{x}_i^{\mathtt{pq}} - (\hat{\mathtt{x}}_j^{\mathtt{pq}} - \hat{\mathtt{x}}_0^{\mathtt{pq}}) \equiv_M h^{\mathtt{p}}_{ij} \wedge \bigwedge_{(i,j)\in J^{\mathtt{pq}}} \mathtt{x}_i^{\mathtt{pq}} - \mathtt{x}_j^{\mathtt{pq}} \equiv_M h^{\mathtt{pq}}_{ij} \wedge \bigwedge_{(i,j)\in J^{\mathtt{q}}} (\mathtt{x}_i^{\mathtt{pq}} + \hat{\mathtt{x}}_0^{\mathtt{pq}}) - \mathtt{x}_j^{\mathtt{q}} \equiv_M h^{\mathtt{q}}_{ij} \\
\psi^{\mathtt{q}}_{\bar{\mathtt{y}}^{\mathtt{pq}}} \equiv{}& \bigwedge_{(i,j)\in K^{\mathtt{p}}} \mathtt{y}_i^{\mathtt{pq}} \approx^{\mathtt{p}}_{ij} \hat{\mathtt{y}}_j^{\mathtt{pq}} \ominus \hat{\mathtt{y}}_0^{\mathtt{pq}} \wedge \bigwedge_{(i,j)\in K^{\mathtt{pq}}} \mathtt{y}_i^{\mathtt{pq}} \approx^{\mathtt{pq}}_{ij} \mathtt{y}_j^{\mathtt{pq}} \wedge \bigwedge_{(i,j)\in K^{\mathtt{q}}} \mathtt{y}_i^{\mathtt{pq}} \oplus \hat{\mathtt{y}}_0^{\mathtt{pq}} \approx^{\mathtt{q}}_{ij} \mathtt{y}_j^{\mathtt{q}}.
\end{aligned}$$

Important point: Quantifier elimination is done *by hand*, since we need an equivalent *constraint* (not an arbitrary quantifier-free formula).

Desynchronised semantics

Useful technique for the analysis of TCA [Pachl'82; Krčál,Yi'06].
Main idea:

- Allow processes to elapse time *locally*: $\Delta(p,q) \geq 0$ if $p \Rightarrow q$.
  - Receivers are allowed to be ahead of senders, but not vice versa.
  - This preserve causality of message receptions.
- Messages $p \Rightarrow q$ have their age increased by $\Delta(p,q)$.
- Weaker semantics (more runs).

What do we gain?

- By scheduling senders far enough in the future,
  we can keep the channels empty → Rendezvous semantics.

Rendezvous semantics

Useful technique for the analysis of TCA [Pachl'82; Krčál,Yi'06].
Main idea:

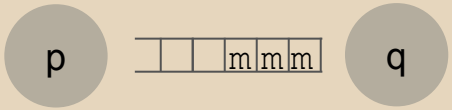
- Execute simultaneously $!m$ with its matching $?m$.
- Stronger semantics (less runs).

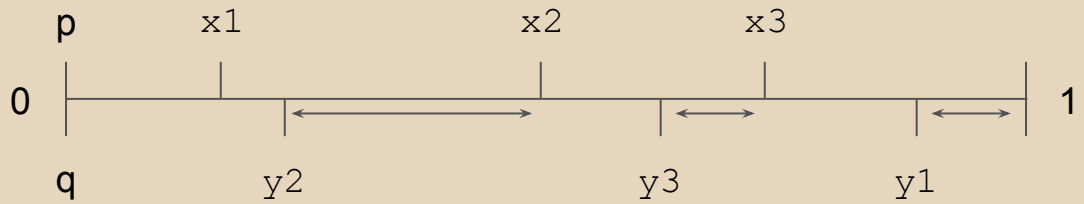
Lemma. Over *polyforest topologies*, the standard semantics is equivalent to the desynchronised+rendezvous semantics.

How to measure the desynchronisation?

- Integral part: Add a $\mathbb{N}$-counter for each receiver.
- Fractional part: Cyclic order atoms.

The issue with fractional values





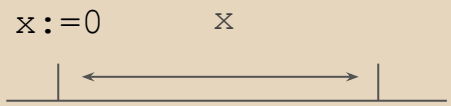
Suppose we advance the time of process q.

- It is not sufficient to keep track of a global region for clocks of p and q.
- We need to keep track also of the total order of *differences* `xi - yj`.
- Two ways to solve this:
  - Clock difference relations `x - y ~ z - t, x - y ~ 1 - (z - t).`
  - Cyclic order atoms (only reference points move).

From clocks to registers

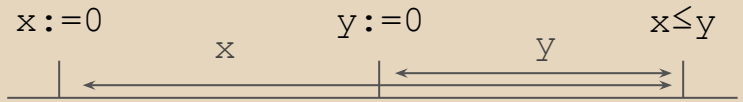
- A special register `now` stores the current time.
- For each clock `x` there is a register `x'` storing the value of `now` at the time of the last reset of `x`.

clocks

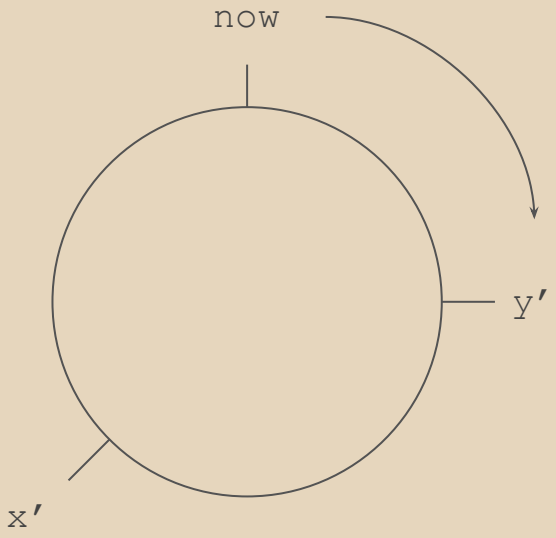


registers `x':=now` `x = now⊖x'`

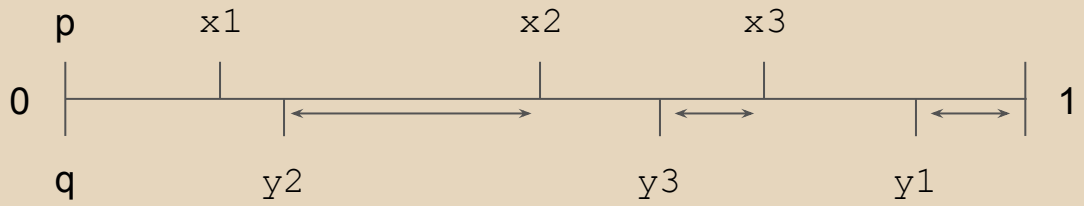
clocks



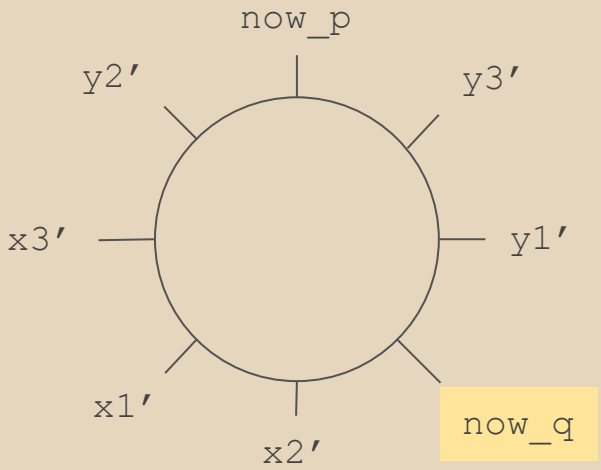
registers `x':=now` `y':=now` `K(now,y',x') ∨ now=x' ∨ y'=x'`



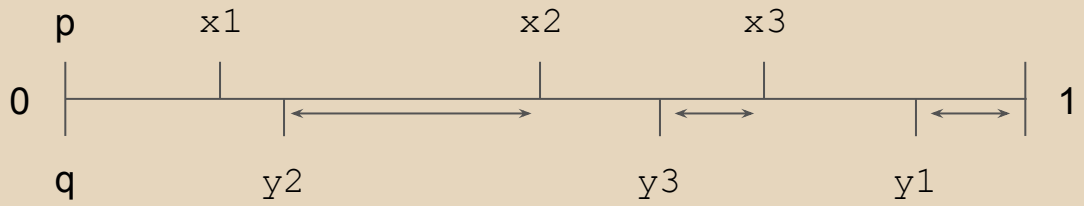
From clocks to registers: time elapse



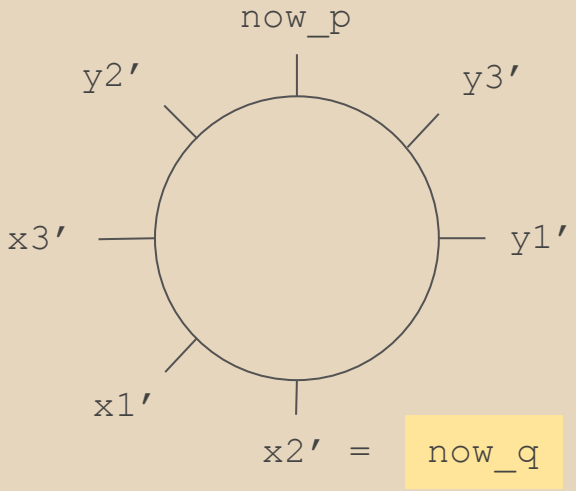
Advance the time q:



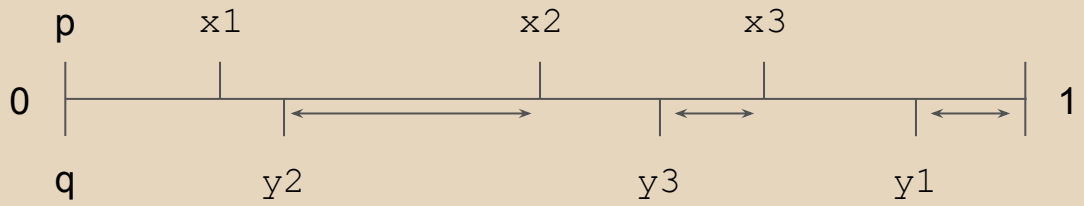
From clocks to registers: time elapse



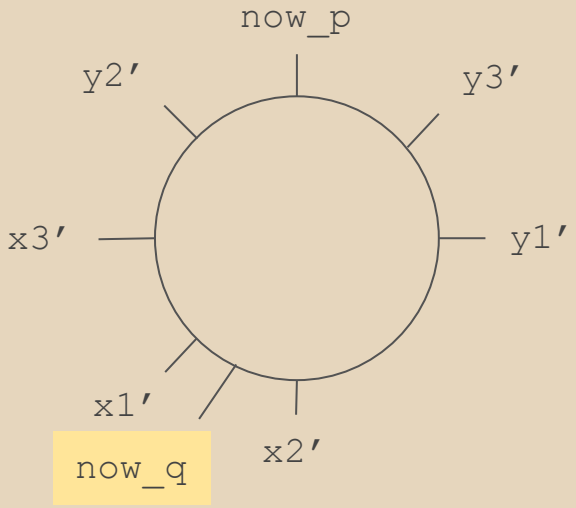
Advance the time q:



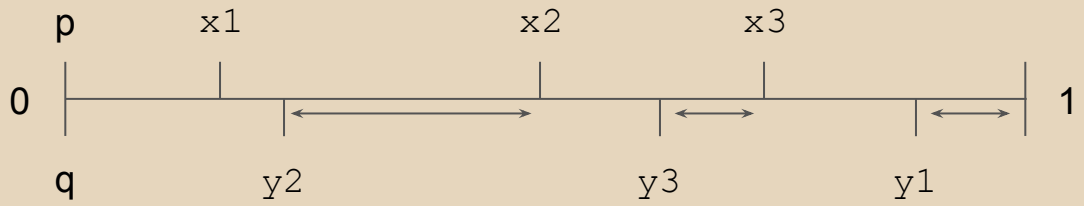
From clocks to registers: time elapse



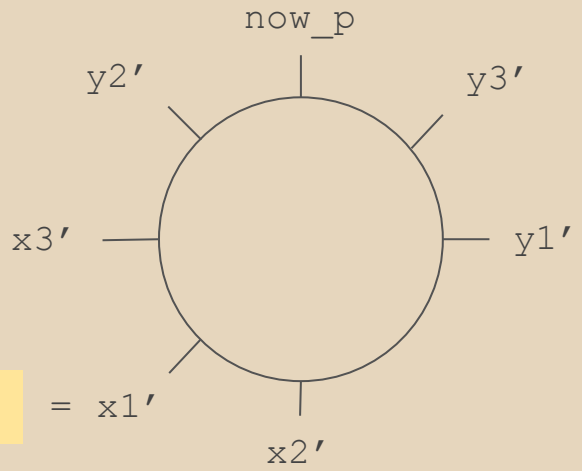
Advance the time q:



From clocks to registers: time elapse



Advance the time q:



now_q = x1′

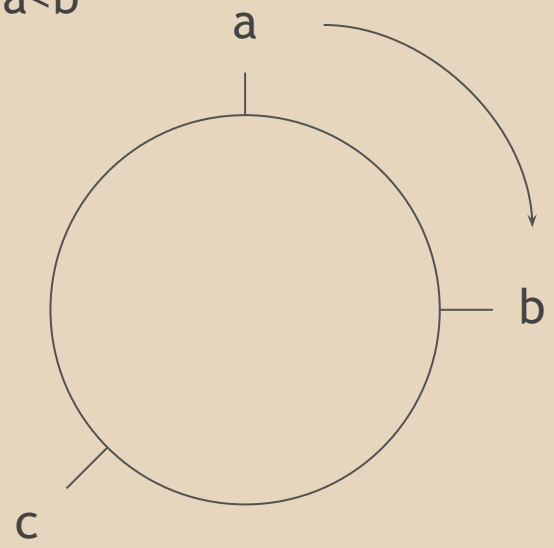
Cyclic order atoms

Consider the structure ([0, 1), K), where $K \subseteq \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ is defined as

$$K(a, b, c) \leftrightarrow\square\ a<b<c \vee b<c<a \vee c<a<b$$

Important properties of cyclic order atoms:

- Satisfiability is decidable.
- Effective elimination of quantifiers.
  - Register constraints.
- Homogeneous (finitely many regions).



Register automata with counters (RAC)

Simulate the desynchronised+rendezvous semantics of a simple TCA with a *register automaton with $\mathbb{N}$-counters*:

- For every channel $p \Rightarrow q$ there is a counter c measuring the integral desynchronisation between p and q.
  - Counters are 0 at the beginning and at the end of the simulation.
  - Counters can be incremented and decremented by 1.
  - Simple send $x=0$ and matching receive $x \sim k$ are simulated by $c \sim k$.
    - Test for zero only if $p \Rightarrow q$ has inequality tests.
- For each local clock x there is a register over cyclic order atoms storing the fractional part now of the last time x was reset.
  - Fractional clock constraint → register constraints.

Summary

1. Reduce to the more constrained *simple TCA*:
   a. The initial value of message clock(s) is `0`.
   b. Reception constraints are either
      i. Integral non-diagonal: `x ~ k`, or
      ii. Fractional equality: `{y} = {z}`.

   Achieved via the method of *quantifier elimination*.
2. Desynchronised semantics (receivers ahead of senders).
3. Rendezvous semantics (handshaking communication → no channels).
4. Simulate 2,3 with *register automata with counters* (RAC).
   a. Counters keep track of the integral desynchronisation.
   b. Registers keep track of fractional values.

Further directions

- Are channel languages of polyforest topologies timed regular?
- Decidable subclasses of integer inequality constraints
  - Upward closed constraints $z \geq k$.
- Finer notions of communication topologies.
  - Take into account the local control structure.
- Application to multiparty session types?
- More general data:
  - What are the conditions on data preserving decidability?