Verification of Input-bounded FIFO Machines

Amrita Suresh
Supervised by Prof. Alain Finkel

Laboratoire Spécification et Vérification, ENS Paris-Saclay

March-July 2019

Special thanks to Dr. Benedikt Bollig

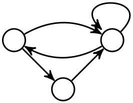
Table of Contents

FIFO Machines

Distributed processes such that

- each process is a finite state machine

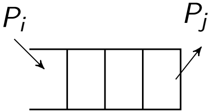
P_i :



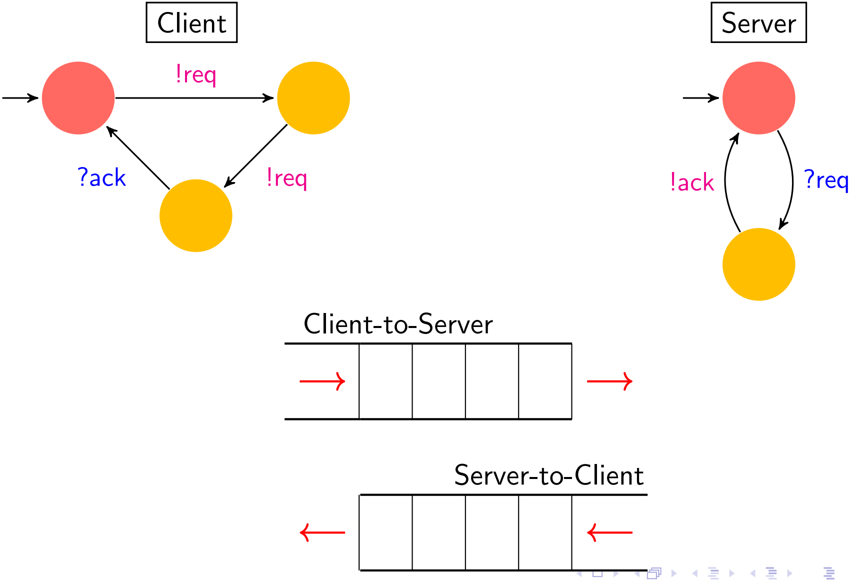
- there are a fixed number of processes

$P_1, \dots, P_n$

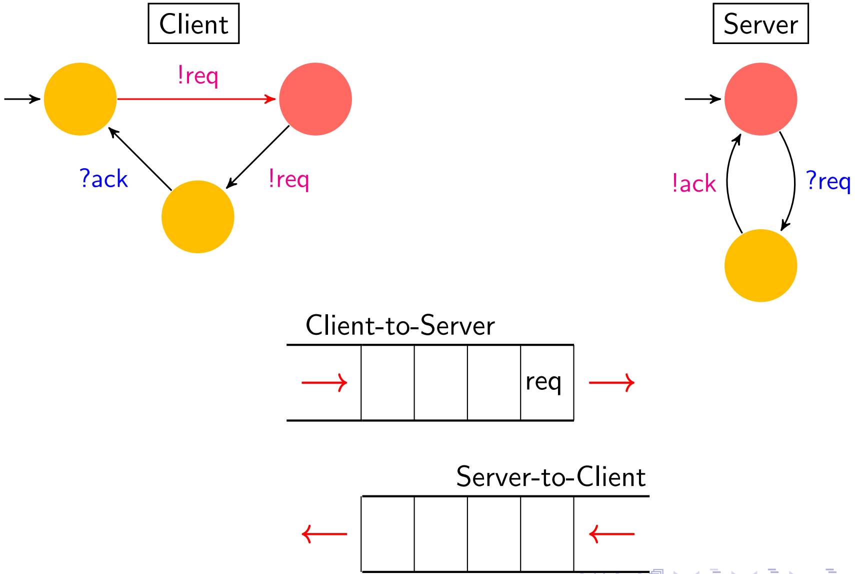
- they communicate using queues



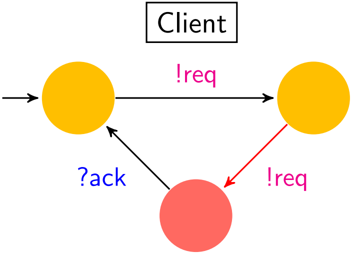
Example

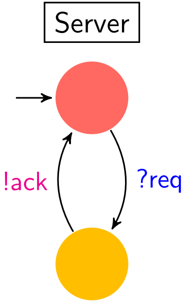

Client
!req
?ack
!req
Server
!ack
?req
Client-to-Server
Server-to-Client

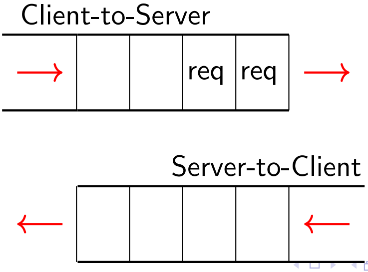
Example


Client
!req
?ack
!req
Server
!ack
?req
Client-to-Server
req
Server-to-Client

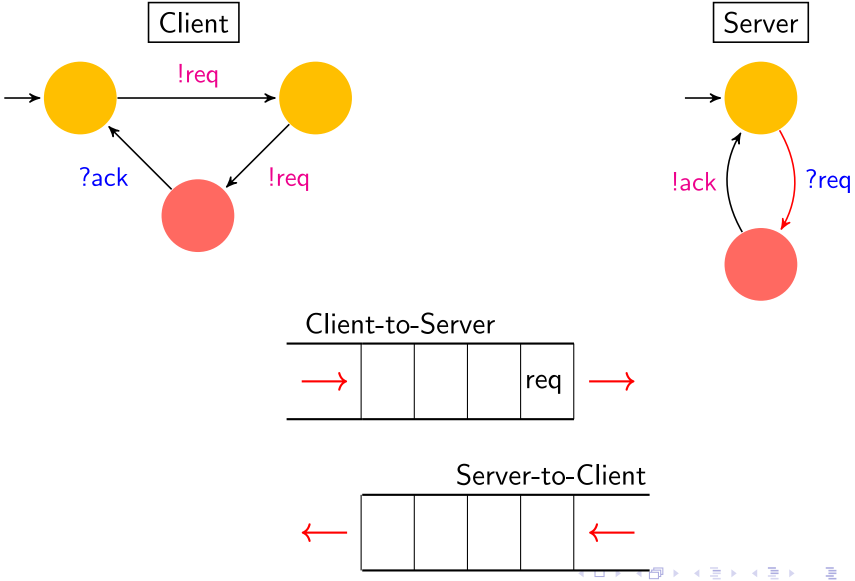
Example


Client
!req
!req
?ack

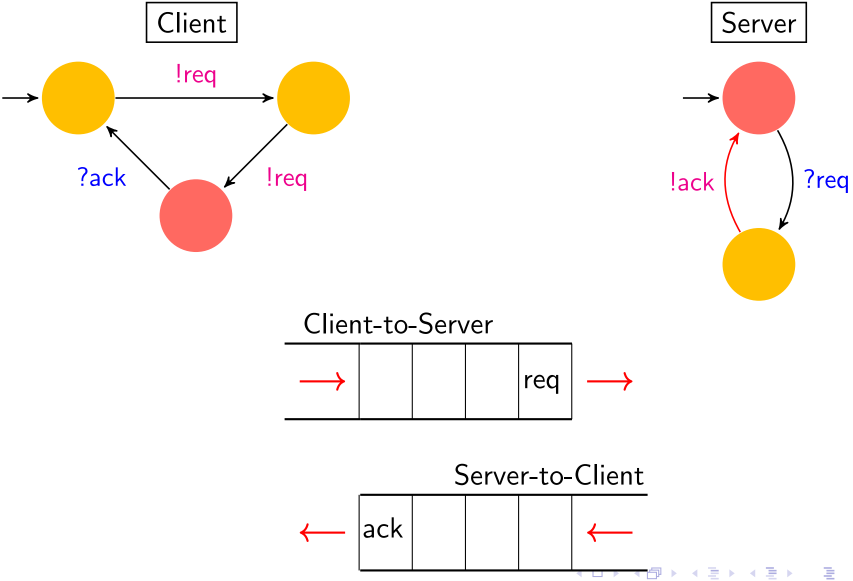

Server
!ack
?req


Client-to-Server
req
req
Server-to-Client

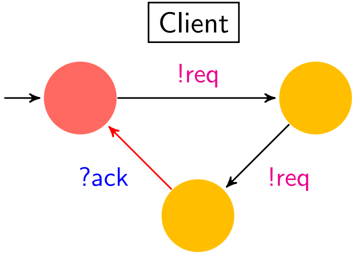
Example

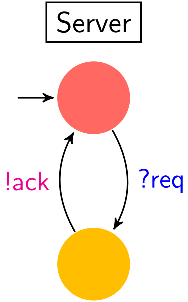

Client
!req
?ack
!req
Server
!ack
?req
Client-to-Server
req
Server-to-Client

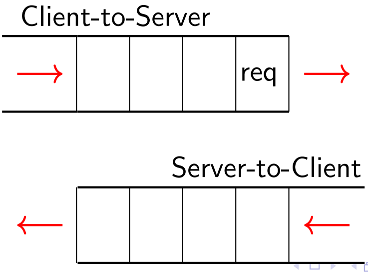
Example


Client
!req
?ack
!req
Server
!ack
?req
Client-to-Server
req
Server-to-Client
ack

Example


Client
!req
?ack
!req


Server
!ack
?req


Client-to-Server
req
Server-to-Client

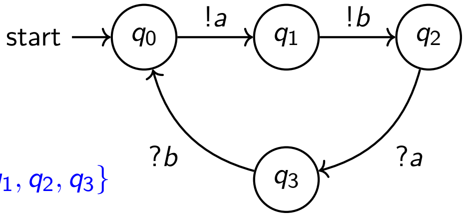
Formal definition

A FIFO machine $\mathcal{S}$ with one channel c is defined as $\mathcal{S} = (Q, \Sigma, T)$ where

- Q is a finite set of control-states,
- Σ is the alphabet,
- $T \subseteq Q \times \{!, ?\} \times \Sigma \times Q$ is the transition relation.

The system is said to be in a configuration $s = (q, w)$ when the control-state is q and the contents of the channel are w.

Formal definition



$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b\}$

Figure: A FIFO system $\mathcal{S}$ with initial configuration (q_0, ϵ)

.

Verification problems

Give a FIFO machine $\mathcal{S}$ with an initial configuration $s_0 = (q_0, w_0)$,

- $\mathcal{S}$ terminates if it has no infinite run.

Verification problems

Give a FIFO machine $\mathcal{S}$ with an initial configuration $s_0 = (q_0, w_0)$,

- $\mathcal{S}$ terminates if it has no infinite run.
- $\mathcal{S}$ is bounded if $Post^*(s_0)$ is finite.

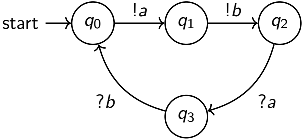
Verification problems

Give a FIFO machine $\mathcal{S}$ with an initial configuration $s_0 = (q_0, w_0)$,

- $\mathcal{S}$ terminates if it has no infinite run.
- $\mathcal{S}$ is bounded if $Post^*(s_0)$ is finite.

Note

Termination implies boundedness, but the converse is false.



$(q_0, \epsilon) \rightarrow (q_1, a) \rightarrow (q_2, ab) \rightarrow (q_3, b) \rightarrow (q_0, \epsilon) \rightarrow \ldots$ is a non-terminating run.
$Post^*(s_0)$ is bounded.

Verification problems

Give a FIFO machine $\mathcal{S}$ with an initial configuration $s_0 = (q_0, w_0)$,

- $\mathcal{S}$ terminates if it has no infinite run.
- $\mathcal{S}$ is bounded if $Post^*(s_0)$ is finite.

Theorem

Testing the unboundedness of a channel in a general FIFO system is undecidable.[1]

[1] Daniel Brand and Pitro Zafiropulo (1983). "On communicating finite-state machines". In: *Journal of the ACM (JACM)* 30.2, pp. 323–342.

Verification problems

Give a FIFO machine $\mathcal{S}$ with an initial configuration $s_0 = (q_0, w_0)$,

- $\mathcal{S}$ terminates if it has no infinite run.
- $\mathcal{S}$ is bounded if $Post^*(s_0)$ is finite.
- a configuration (q, w) is reachable if $\exists \sigma$ such that $(q_0, w_0) \xrightarrow{\sigma} (q, w)$.

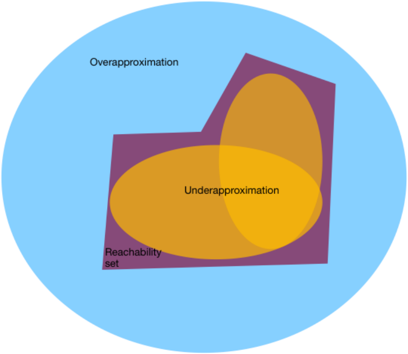
Verification problems

Give a FIFO machine $\mathcal{S}$ with an initial configuration $s_0 = (q_0, w_0)$,

- $\mathcal{S}$ terminates if it has no infinite run.
- $\mathcal{S}$ is bounded if $Post^*(s_0)$ is finite.
- a configuration (q, w) is reachable if $\exists \sigma$ such that $(q_0, w_0) \xrightarrow{\sigma} (q, w)$.
- a control-state q is reachable if $\exists \sigma$ and $\exists w$ a channel valuation such that $(q_0, w_0) \xrightarrow{\sigma} (q, w)$.

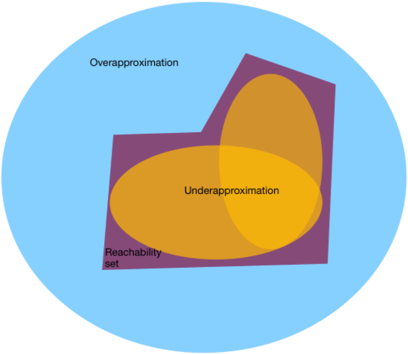
Approach to verification

- Idea: to use over and under-approximations for verification



Approach to verification

- Idea: to use over and under-approximations for verification



- "Input-bounded FIFO systems" - an underapproximation

Some subclasses of FIFO systems

The following subclasses have decidable properties.

- *Half-duplex systems* with two processes (but extension to three processes leads to undecidability). (Cécé and Finkel 2005)
- *Lossy FIFO systems*. (Abdulla et al. 2004)
- *Existentially-bounded deadlock-free FIFO automata**. (Genest, Kuske, and Muscholl 2007)
- *Synchronisable FIFO systems**. (Alain Finkel and Lozes 2017)
- *Flat FIFO systems* (most verification problems are in *NP*). (Alain Finkel and Praveen 2019)

Input-bounded FIFO Systems

- *Bounded* language - $L \subseteq w_1^* \ldots w_k^*$.
- *Input*-bounded - L_{send} is bounded.

Input-bounded FIFO Systems

- *Bounded* language - $L \subseteq w_1^* \ldots w_k^*$.
- *Input*-bounded - L_{send} is bounded.
- *Reachability*-bounded - Channel contents belong to a bounded language.

Input-bounded FIFO Systems

- *Bounded* language - $L \subseteq w_1^* \dots w_k^*$.
- *Input*-bounded - L_{send} is bounded.
- *Reachability*-bounded - Channel contents belong to a bounded language.

Theorem

A two-counter Minsky machine can be simulated by a reachability-bounded FIFO system.

Table of Contents

Well Structured Transition Systems (A. Finkel and Schnoebelen 2001)

- A wqo over a set $X \implies$ every infinite sequence $x_0, x_1, x_2, \ldots$ over X contains an increasing pair:
  $\exists i < j$ s.t. $x_i \leq x_j$.

Well Structured Transition Systems (A. Finkel and Schnoebelen 2001)

- A wqo over a set $X \implies$ every infinite sequence $x_0, x_1, x_2, \ldots$ over X contains an increasing pair:
  $\exists i < j$ s.t. $x_i \leq x_j$.
- Example
  - $\mathbb{N}$ over the ordering $\leq$ is wqo.
  - $\mathbb{Z}$ over the ordering $\leq$ is not. e.g. $-1 \geq -2 \geq -3 \ldots$ has no increasing pair.

Well Structured Transition Systems (A. Finkel and Schnoebelen 2001)

- A wqo over a set $X \implies$ every infinite sequence $x_0, x_1, x_2, \ldots$ over X contains an increasing pair:
  $\exists i < j$ s.t. $x_i \leq x_j$.
- The transition system $(X, \to)$ has strong compatibility i.e.

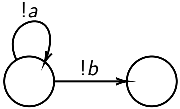
$$\begin{tikzcd} s \arrow[r, phantom, "\lll"] \arrow[d, "\delta"'] & t \arrow[d, "\delta"] \\ s' \arrow[r, phantom, "\lll"] & t' \end{tikzcd}$$

Branch-WSTS

- $\mathcal{S} = (X, \rightarrow, \leq)$ is *branch-WSTS* if it is
- *branch-wqo*
  - if for every infinite run $n_0(x_0) \rightarrow n_1(x_1) \rightarrow n_2(x_2), \ldots$ of $\mathcal{S}$, $x_0, x_1, \ldots$ is wqo.

Branch-WSTS

- $\mathcal{S} = (X, \rightarrow, \leq)$ is *branch-WSTS* if it is
- *branch-wqo*
  - if for every infinite run $n_0(x_0) \rightarrow n_1(x_1) \rightarrow n_2(x_2), \ldots$ of $\mathcal{S}$, $x_0, x_1, \ldots$ is wqo.



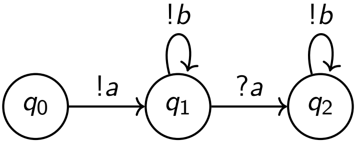
This FIFO system is branch-wqo under the prefix ordering but it is not wqo.

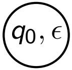
Branch-WSTS

- $\mathcal{S} = (X, \rightarrow, \leq)$ is *branch-WSTS* if it is
- *branch-wqo*
- *branch-compatible*
  - if for all configurations s, t, s' such that $s \preceq t$ and $s \xrightarrow{a} s' \xrightarrow{w} t$ implies that there exists a t' such that $t \xrightarrow{a} t'$ and $s' \preceq t'$.

$$\begin{tikzcd}
s \arrow[r, phantom, "\ll"] \arrow[d, "a"'] & t \arrow[d, "a"] \\
s' \arrow[ur, "w"] \arrow[r, phantom, "\ll"] & t'
\end{tikzcd}$$

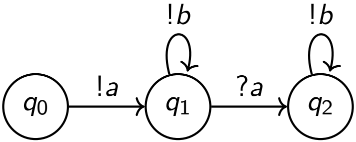
Finite Reachability Tree[1]

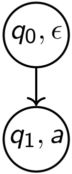




[1] Adapted from A. Finkel and Ph. Schnoebelen (2001). "Well-structured transition systems everywhere!". In: *Theoretical Computer Science* 256.1, pp. 63–92

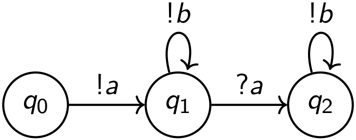
Finite Reachability Tree[1]

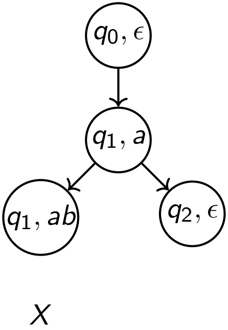




[1] Adapted from A. Finkel and Ph. Schnoebelen (2001). "Well-structured transition systems everywhere!". In: *Theoretical Computer Science* 256.1, pp. 63–92

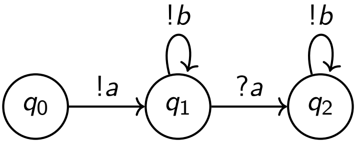
Finite Reachability Tree[1]

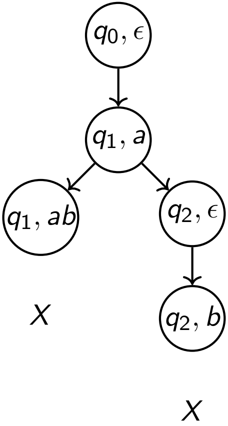




[1] Adapted from A. Finkel and Ph. Schnoebelen (2001). "Well-structured transition systems everywhere!". In: *Theoretical Computer Science* 256.1, pp. 63–92

Finite Reachability Tree[1]





[1]Adapted from A. Finkel and Ph. Schnoebelen (2001). "Well-structured transition systems everywhere!". In: *Theoretical Computer Science* 256.1, pp. 63–92

Why branch-WSTS?

Theorem

Boundedness and termination are decidable for branch-WSTS, if $\leq$ is a decidable, partial ordering, and has computable successor.

Why branch-WSTS?

Theorem

Boundedness and termination are decidable for branch-WSTS, if $\leq$ is a decidable, partial ordering, and has computable successor.

Proof sketch

- A branch-WSTS $\mathcal{S} = (S, \rightarrow, \leq)$ has a finite reachability tree.
- Unbounded iff there exist two configurations in the finite reachability tree such that $s_1 \xrightarrow{*} s_2$ and $s_1 < s_2$.
- Non-terminating iff there exists a subsumed node in the FRT.

Input-bounded FIFO systems over the prefix-ordering

Theorem

Input-bounded FIFO automata are branch-wqo for the prefix-ordering $\leq_{pref}$.

But they are not branch-compatible.

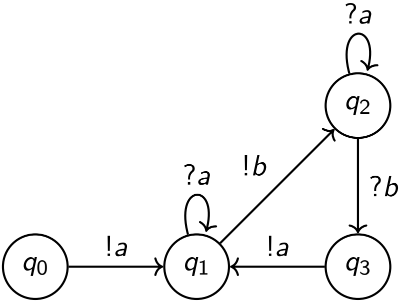


Figure: Consider $\mathcal{S}$, and configurations (q_1, ϵ) and (q_1, a)

Prefix compatible relation

For two configurations $s = (q, w), s' = (q', w')$, $(q, w) \preceq_{comp} (q', w')$ if

- $(q, w) \leq_{pref} (q', w')$ and
- $\exists \sigma$ (with send and receive actions y_σ and x_σ resp.) such that $s \xrightarrow{\sigma} s'$ and
- $x_\sigma = \epsilon$ or
- $|x_\sigma| \leq |y_\sigma|$ and $x_\sigma^\omega = w.y_\sigma^\omega$).

Theorem

FIFO systems are branch-compatible for the relation $\preceq_{comp}$.

Termination

- The prefix compatible relation is not an ordering.

Theorem

Under this relation, we can construct a finite reachability tree for input-bounded FIFO systems.[2]

[2] Thierry Jéron and Claude Jard (1993). "Testing for Unboundedness of FIFO Channels.". In: *Theor. Comput. Sci.* 113, pp. 93–117

Termination

Theorem

Termination is decidable for input-bounded FIFO systems

Table of Contents

Reachability results

Theorem

Reachability and control-state reachability are reducible to one another for input-bounded FIFO systems.

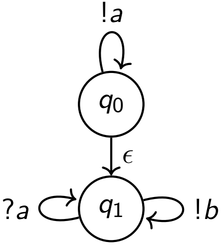
Reachability of input-letter bounded systems

Theorem

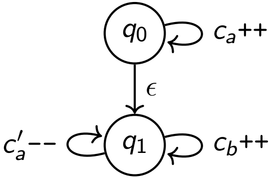
Control state reachability for input-letter bounded FIFO systems is decidable.

Proof idea

Input-letter bounded FIFO systems can be simulated by counter machines with hierarchical zero tests.

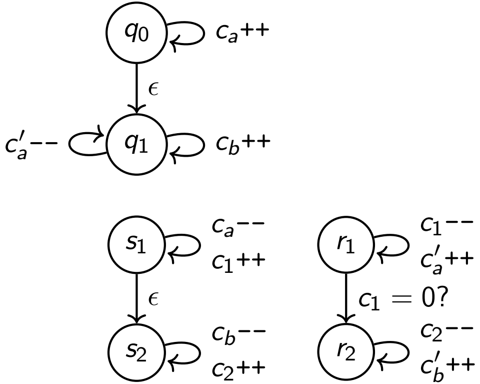


A FIFO machine $\mathcal{S}$.



Proof idea

Input-letter bounded FIFO systems can be simulated by counter machines with hierarchical zero tests.



Counter automata corresponding to the FIFO machine $\mathcal{S}$.

Conclusion

System	Boundedness	Termination	Reachability
General FIFO systems	No [B83]		
Lossy Channel systems	Yes [A04]	Yes	Yes
Flat Systems	Yes [F19]	Yes	Yes
Reachability Bounded systems	No		
Input-bounded systems	Yes [J93]	Yes	?
Input-different-letter bounded systems	Yes	Yes	Yes [F87]
Input-letter bounded systems	Yes	Yes	Yes

Table: Verification problems

Thank you!
Questions?

Some additional information

- The half-duplex property for two machines and two channels (one in each direction) says that each reachable configuration has at most one channel non-empty.

Abdulla, Parosh Aziz et al. (2004). "Using forward reachability analysis for verification of lossy channel systems". In: *Formal Methods in System Design* 25.1, pp. 39–65.

Brand, Daniel and Pitro Zafiropulo (1983). "On communicating finite-state machines". In: *Journal of the ACM (JACM)* 30.2, pp. 323–342.

Cécé, Gérard and Alain Finkel (2005). "Verification of programs with half-duplex communication". In: *Information and Computation* 202.2, pp. 166–190.

Choquet, A and A Finkel (1987). "Simulation of linear FIFO nets having a structured set of terminal markings". In:

Finkel, Alain and Etienne Lozes (2017). "Synchronizability of communicating finite state machines is not decidable". In: *ICALP*.

Finkel, Alain and M Praveen (2019). "Verification of Flat FIFO Systems". In: *CONCUR'19*. Leibniz International Proceedings in Informatics. To appear.

Finkel, A. and Ph. Schnoebelen (2001). "Well-structured transition systems everywhere!". In: *Theoretical Computer Science* 256.1, pp. 63–92.

Genest, Blaise, Dietrich Kuske, and Anca Muscholl (2007). "On communicating automata with bounded channels". In: *Fundamenta Informaticae* 80.1-3, pp. 147–167.

Jéron, Thierry and Claude Jard (1993). "Testing for Unboundedness of FIFO Channels.". In: *Theor. Comput. Sci.* 113, pp. 93–117.