Bounded Reachability Problems are Decidable in FIFO Machines

Benedikt Bollig Alain Finkel **Amrita Suresh**

LSV – CNRS & ENS Paris-Saclay, Université Paris-Saclay, France

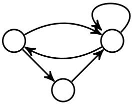
August 16, 2020

FIFO Machines

Distributed processes such that

- Each process P_i is a finite state machine

P_i :

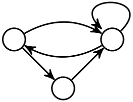


FIFO Machines

Distributed processes such that

- Each process P_i is a finite state machine

P_i :



- There are a fixed number of processes

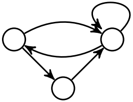
$P_1, ..., P_n$

FIFO Machines

Distributed processes such that

- Each process P_i is a finite state machine

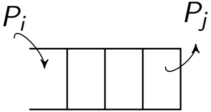
P_i :



- There are a fixed number of processes

$P_1, \ldots, P_n$

- They communicate using FIFO queues



FIFO Machines

- Studied since the 1980s. Widely used in distributed settings.
- FIFO machines simulate TM, hence underapproximations.

[1] Gouda et al., *On deadlock detection in systems of communicating finite state machines*, 1987.
[2] Esparza et al., *Perfect Model for Bounded Verification*, 2012
[3] Finkel and Praveen, *Verification of Flat FIFO Systems*, 2019

FIFO Machines

- Studied since the 1980s. Widely used in distributed settings.
- FIFO machines simulate TM, hence underapproximations.
- Letter-bounded FIFO machines. [1]

[1] Gouda et al., *On deadlock detection in systems of communicating finite state machines*, 1987.

[2] Esparza et al., *Perfect Model for Bounded Verification*, 2012

[3] Finkel and Praveen, *Verification of Flat FIFO Systems*, 2019

FIFO Machines

- Studied since the 1980s. Widely used in distributed settings.
- FIFO machines simulate TM, hence underapproximations.
- Letter-bounded FIFO machines. [1]
- Flat FIFO systems. [2] [3]

[1] Gouda et al., *On deadlock detection in systems of communicating finite state machines*, 1987.

[2] Esparza et al., *Perfect Model for Bounded Verification*, 2012

[3] Finkel and Praveen, *Verification of Flat FIFO Systems*, 2019

FIFO Machines

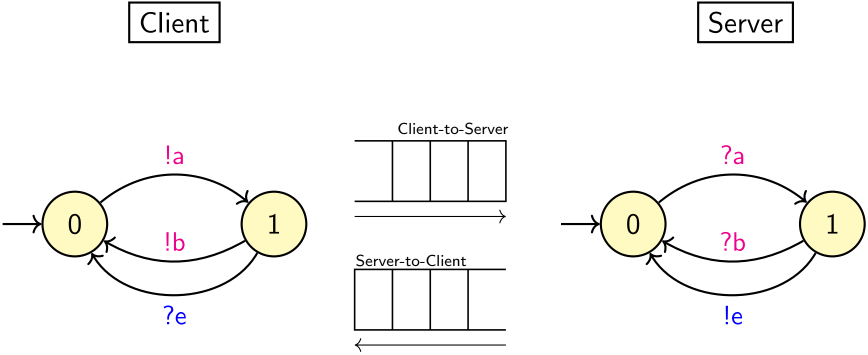
- Studied since the 1980s. Widely used in distributed settings.
- FIFO machines simulate TM, hence underapproximations.
- Letter-bounded FIFO machines. [1]
- Flat FIFO systems. [2] [3]
- (Input-)Bounded FIFO machines strictly contain these subclasses.

[1] Gouda et al., *On deadlock detection in systems of communicating finite state machines*, 1987.

[2] Esparza et al., *Perfect Model for Bounded Verification*, 2012

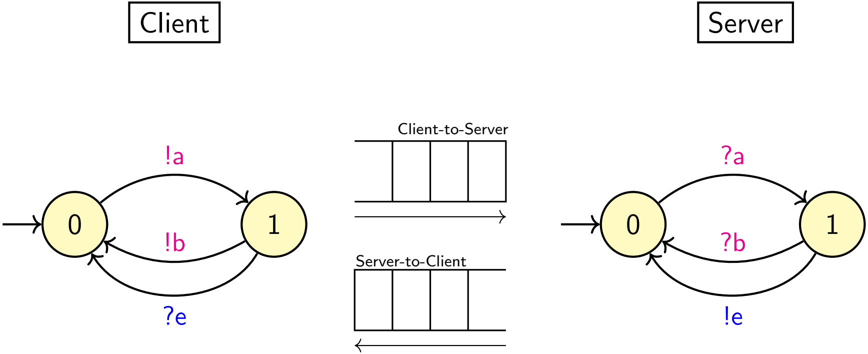
[3] Finkel and Praveen, *Verification of Flat FIFO Systems*, 2019

Example (Connection-Deconnection Protocol) [4]



[4] Jéron, *Testing for unboundedness of FIFO channels*, 1991.

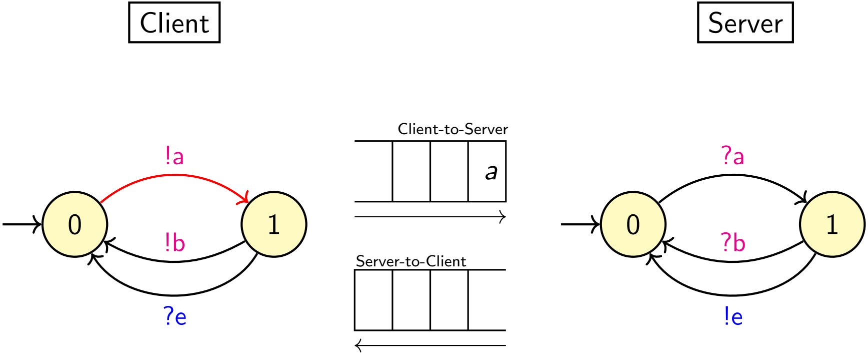
Example (Connection-Deconnection Protocol) [4]



Initial configuration $(0, 0; \varepsilon, \varepsilon)$

[4] Jéron, *Testing for unboundedness of FIFO channels*, 1991.

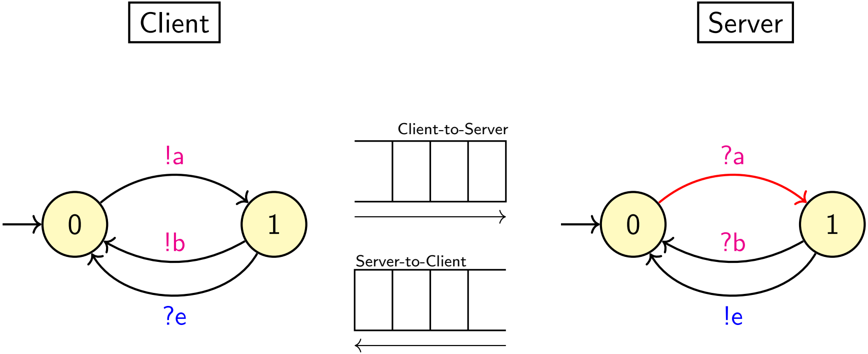
Example (Connection-Deconnection Protocol) [4]



Run - $(0, 0; \varepsilon, \varepsilon) \xrightarrow{!a} (1, 0; a, \varepsilon)$

[4] Jéron, *Testing for unboundedness of FIFO channels*, 1991.

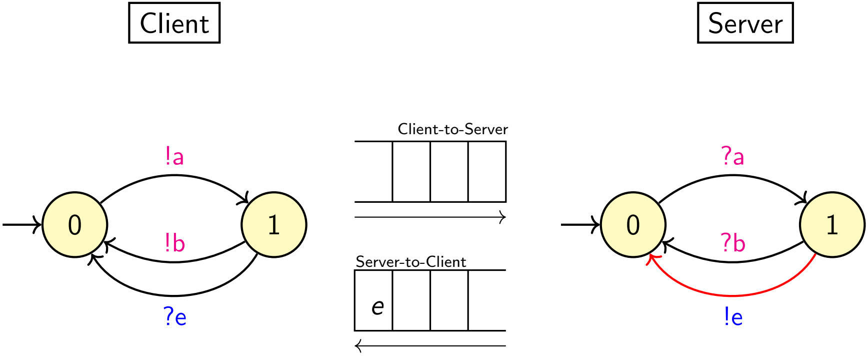
Example (Connection-Deconnection Protocol) [4]



$$(0, 0; \varepsilon, \varepsilon) \xrightarrow{!a} (1, 0; a, \varepsilon) \xrightarrow{?a} (1, 1; \varepsilon, \varepsilon)$$

[4] Jéron, *Testing for unboundedness of FIFO channels*, 1991.

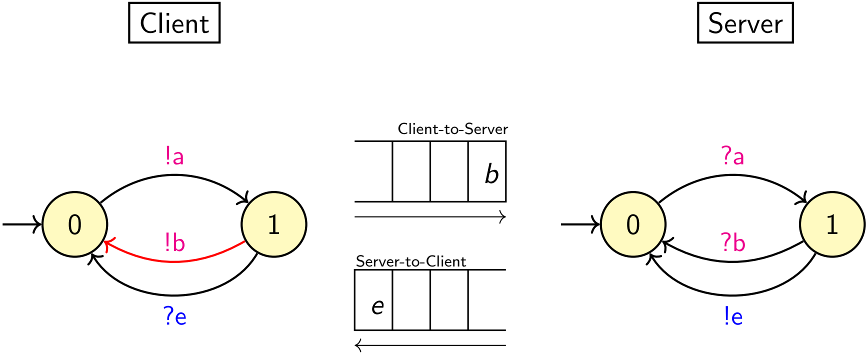
Example (Connection-Deconnection Protocol) [4]



$(0, 0; \varepsilon, \varepsilon) \xrightarrow{!a} (1, 0; a, \varepsilon) \xrightarrow{?a} (1, 1; \varepsilon, \varepsilon) \xrightarrow{!e} (1, 0; \varepsilon, e)$

[4] Jéron, *Testing for unboundedness of FIFO channels*, 1991.

Example (Connection-Deconnection Protocol) [4]

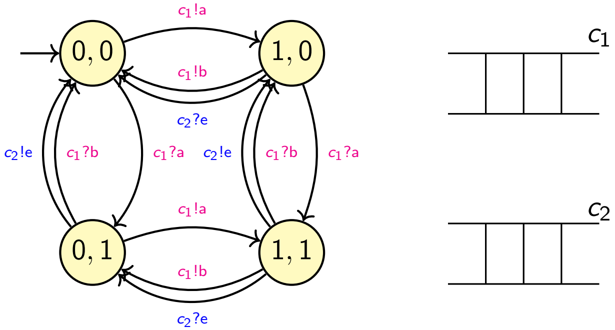


$$(0,0;\varepsilon,\varepsilon) \xrightarrow{!a} (1,0;a,\varepsilon) \xrightarrow{?a} (1,1;\varepsilon,\varepsilon) \xrightarrow{!e} (1,0;\varepsilon,e) \xrightarrow{!b} (0,0;b,e)$$

[4] Jéron, *Testing for unboundedness of FIFO channels*, 1991.

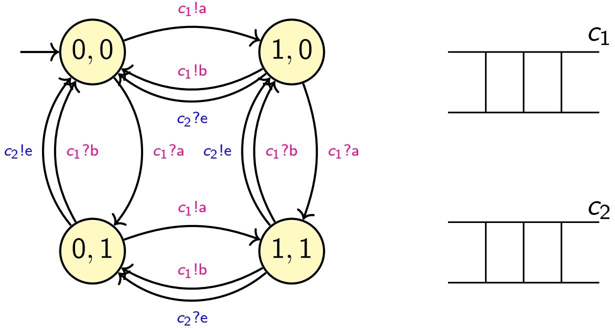
Formal model

A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where



Formal model

A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where

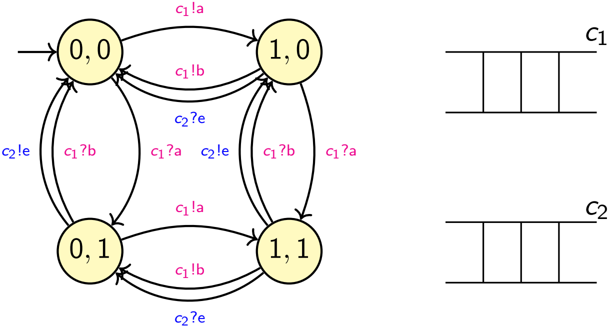


Q is a finite set of control-states.

$Q = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$.

Formal model

A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where

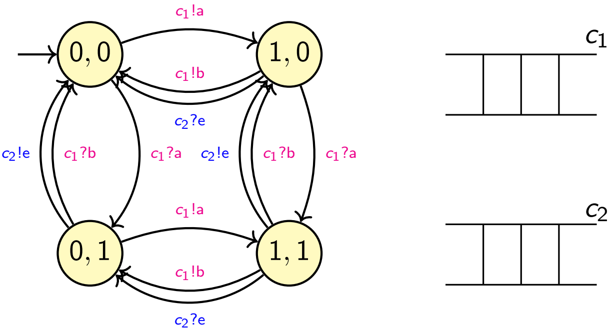


Ch is the number of channels.

$Ch = \{c_1, c_2\}$.

Formal model

A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where

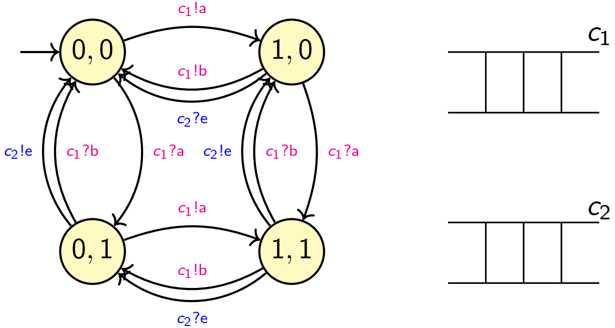


$\Sigma = \uplus_{c \in Ch} \Sigma_c$ is the alphabet.

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

Formal model

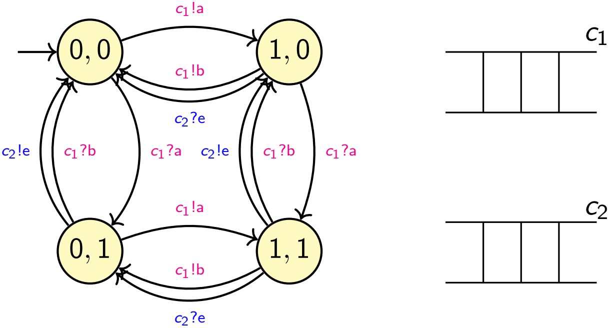
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where



$T \subseteq Q \times A_M \times Q$ is the transition relation

Formal model

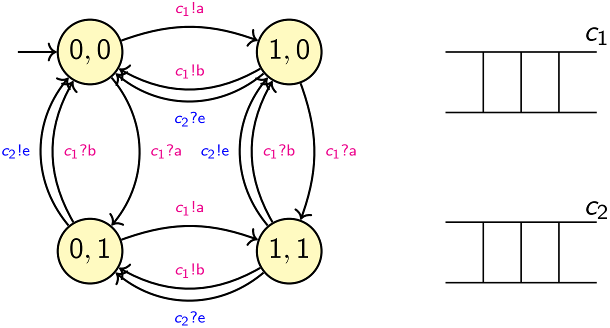
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where



$T \subseteq Q \times A_M \times Q$ is the transition relation where
$A_M = \{c!a \mid a \in \Sigma \text{ and } c \in Ch\} \cup \{c?a \mid a \in \Sigma \text{ and } c \in Ch\}$

Formal model

A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where



q_0 is the initial state.

$q_0 = (0, 0)$.

Configurations and Reachability

- A configuration is $(q, \mathbf{w})$ where q is the control-state and $\mathbf{w}$ is a tuple of the channel contents. The set of configurations is S_M.

Configurations and Reachability

- A configuration is $(q, \mathbf{w})$ where q is the control-state and $\mathbf{w}$ is a tuple of the channel contents. The set of configurations is S_M.
- $Reach_M = \{s \in S_M \mid (q_0, \varepsilon) \xrightarrow{\sigma} s \text{ for some } \sigma \in A_M^*\}$.

Configurations and Reachability

- A configuration is $(q, \mathbf{w})$ where q is the control-state and $\mathbf{w}$ is a tuple of the channel contents. The set of configurations is S_M.
- $Reach_M = \{s \in S_M \mid (q_0, \varepsilon) \xrightarrow{\sigma} s \text{ for some } \sigma \in A_M^*\}$.

Theorem

Testing the reachability of a configuration in a general FIFO system is undecidable. [a]

[a] Brand and Zafiropulo, *On Communicating Finite-State Machines*, 1983.

Configurations and Reachability

- $Reach_M(\sigma) = \{s \in S_M \mid (q_0, \varepsilon) \xrightarrow{\sigma} s\}$ where $\sigma \in A_M^*$.

Configurations and Reachability

- $Reach_M(\sigma) = \{s \in S_M \mid (q_0, \varepsilon) \xrightarrow{\sigma} s\}$ where $\sigma \in A_M^*$.
- $Reach_M(L) = \bigcup_{\sigma \in L} Reach_M(\sigma)$.

Configurations and Reachability

We define the *send projection over c* $proj_{c!} : A_M^* \rightarrow \Sigma^*$

Example: $proj_{c!}(c!x.d!y.c?x.c!z.c!z) = xzz$

Bounded language

Let $w_1, ..., w_n \in \Sigma^+$ be non-empty words where $n \geq 1$.
L is a **bounded language** over $(w_1, ..., w_n)$ if $L \subseteq w_1^* ... w_n^*$.

Bounded language

Let $w_1, ..., w_n \in \Sigma^+$ be non-empty words where $n \geq 1$.
L is a **bounded language** over $(w_1, ..., w_n)$ if $L \subseteq w_1^* ... w_n^*$.

$(ab)^* d(c)^*$ is a bounded language over (ab, d, c).

Bounded language

Let $w_1, ..., w_n \in \Sigma^+$ be non-empty words where $n \geq 1$.
L is a **bounded language** over $(w_1, ..., w_n)$ if $L \subseteq w_1^* ... w_n^*$.

$(ab)^* d(c)^*$ is a bounded language over (ab, d, c).

$((ab)^*(cd)^*)^*$ is not a bounded language.

Bounded language

Let $w_1, ..., w_n \in \Sigma^+$ be non-empty words where $n \geq 1$.
L is a **bounded language** over $(w_1, ..., w_n)$ if $L \subseteq w_1^* ... w_n^*$.

Let $L = (L_c)_{c \in Ch}$ be non-empty regular bounded languages over Σ.
$L_! = \{w \in A_M^* \mid proj_{c!}(w) \in L_c \text{ for all } c \in Ch\}$.

Bounded language

Let $w_1, ..., w_n \in \Sigma^+$ be non-empty words where $n \geq 1$.
L is a **bounded language** over $(w_1, ..., w_n)$ if $L \subseteq w_1^* ... w_n^*$.

Let $L = (L_c)_{c \in Ch}$ be non-empty regular bounded languages over Σ.
$L_! = \{w \in A_M^* \mid proj_{c!}(w) \in L_c \text{ for all } c \in Ch\}$.
(We define $L_?$ similarly.)

Rational relations

Let $\mathcal{R} \subseteq \prod_{c \in Ch} \Sigma_c^*$ and $\Theta = \prod_{c \in Ch} (\Sigma_c \cup \varepsilon)$.

Rational relations

Let $\mathcal{R} \subseteq \prod_{c \in Ch} \Sigma_c^*$ and $\Theta = \prod_{c \in Ch}(\Sigma_c \cup \varepsilon)$.
We say that $\mathcal{R}$ is rational if there is a regular word language $R \in \Theta^*$ such that

$$\mathcal{R} = \{(\mathbf{a}_c^1 \cdot \ldots \cdot \mathbf{a}_c^n)_{c \in Ch} \mid \mathbf{a}^1 \ldots \mathbf{a}^n \in R \text{ with } n \in \mathbb{N} \text{ and } \mathbf{a}^i = (\mathbf{a}_c^i)_{c \in Ch} \in \Theta \text{ for } i \in \{1, \ldots, n\}\}.$$

Rational relations

Let $\mathcal{R} \subseteq \prod_{c \in Ch} \Sigma_c^*$ and $\Theta = \prod_{c \in Ch}(\Sigma_c \cup \varepsilon)$.
We say that $\mathcal{R}$ is rational if there is a regular word language $R \in \Theta^*$ such that

$$\mathcal{R} = \{(\mathbf{a}_c^1 \cdot \ldots \cdot \mathbf{a}_c^n)_{c \in Ch} \mid \mathbf{a}^1 \ldots \mathbf{a}^n \in R \text{ with } n \in \mathbb{N} \text{ and } \mathbf{a}^i = (\mathbf{a}_c^i)_{c \in Ch} \in \Theta \text{ for } i \in \{1, \ldots, n\}\}.$$

Example: $\mathcal{R} = \{(a^m, b^n \mid m \geq n)\}$ is a rational relation, witnessed by $R = ((a, b) + (a, \epsilon))^*$.

Input-Bounded Rational Reachability Problem

Given

- a FIFO machine $M = (Q, Ch, \Sigma, T, q_0)$,

Input-Bounded Rational Reachability Problem

Given

- a FIFO machine $M = (Q, Ch, \Sigma, T, q_0)$,
- a control-state $q \in Q$,

Input-Bounded Rational Reachability Problem

Given

- a FIFO machine $M = (Q, Ch, \Sigma, T, q_0)$,
- a control-state $q \in Q$,
- a tuple $L = (L_c)_{c \in Ch}$ of non-empty regular bounded languages over Σ,

Input-Bounded Rational Reachability Problem

Given

- a FIFO machine $M = (Q, Ch, \Sigma, T, q_0)$,
- a control-state $q \in Q$,
- a tuple $L = (L_c)_{c \in Ch}$ of non-empty regular bounded languages over Σ,
- a rational relation $\mathcal{R} \subseteq \prod_{c \in Ch} \Sigma_c^*$.

Input-Bounded Rational Reachability Problem

Given

- a FIFO machine $M = (Q, Ch, \Sigma, T, q_0)$,
- a control-state $q \in Q$,
- a tuple $L = (L_c)_{c \in Ch}$ of non-empty regular bounded languages over Σ,
- a rational relation $\mathcal{R} \subseteq \prod_{c \in Ch} \Sigma_c^*$.

Question: Do we have $(q, \mathbf{w}) \in Reach_M(L_!)$ for some $\mathbf{w} \in \mathcal{R}$?

Input-Bounded Rational Reachability Problem

Theorem

The Input-Bounded Rational Reachability Problem is decidable.

Input-Bounded Rational Reachability Problem

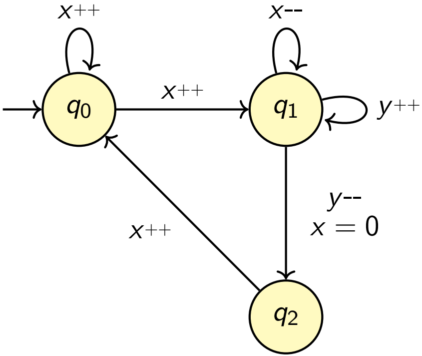
Theorem

The Input-Bounded Rational Reachability Problem is decidable.

Proof using counter machines...

Counter machines

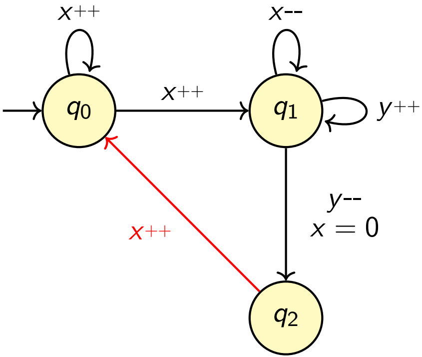
A counter machine (with zero tests) is a tuple $C = (Q, Cnt, T, q_0)$



Counter machines with RESTRICTED zero tests

Once a counter has been tested for zero, it cannot be incremented or decremented anymore.

Counter machines with RESTRICTED zero tests



Counter machines with RESTRICTED zero tests

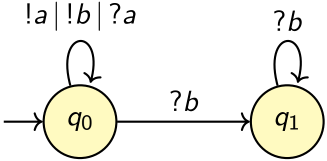
Theorem

The following problem is decidable: Given a counter machine $C = (Q, Cnt, T, q_0)$, a regular language $L \subseteq A_C^*$, a control state $q \in Q$, and counter valuation v, do we have $(q, v) \in Reach_C(L_{zero} \cap L)$?

Translation

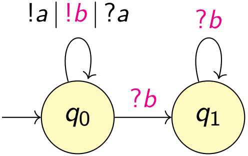
Intuition: Given a bounded language L, which is bounded over $(w_1, \ldots, w_n)$, we construct a counter x_i for each w_i.

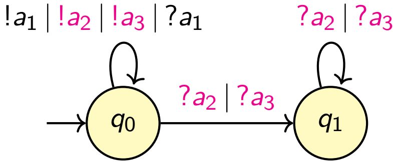
Translation



$$\hat{L} = (ab)^* bb^*$$

Step 1: Distinct letter property

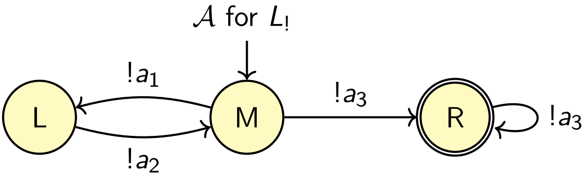




$\hat{L} = (ab)^* bb^*$

$L = (a_1 a_2)^* a_3 a_3{}^*$

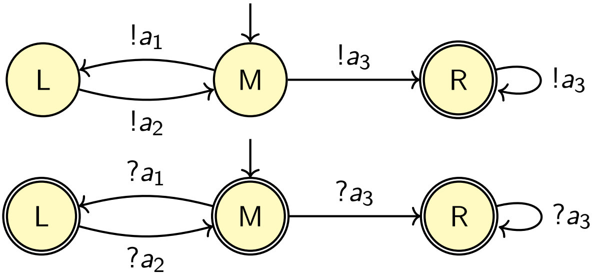
Step 2: Trace property



$$L = (a_1 a_2)^* a_3 a_3^*$$

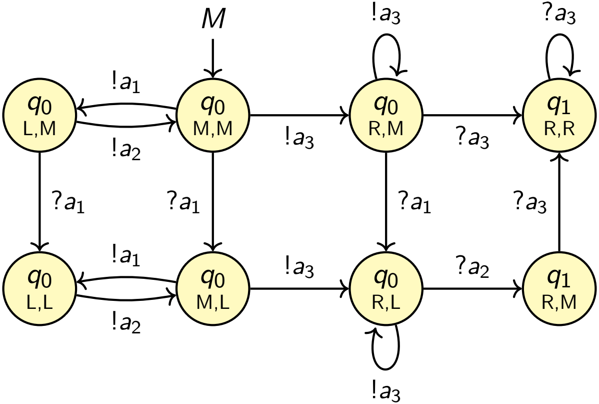
Step 2: Trace property

$\mathcal{A}$ for $L_! \cap Pref(L_?)$

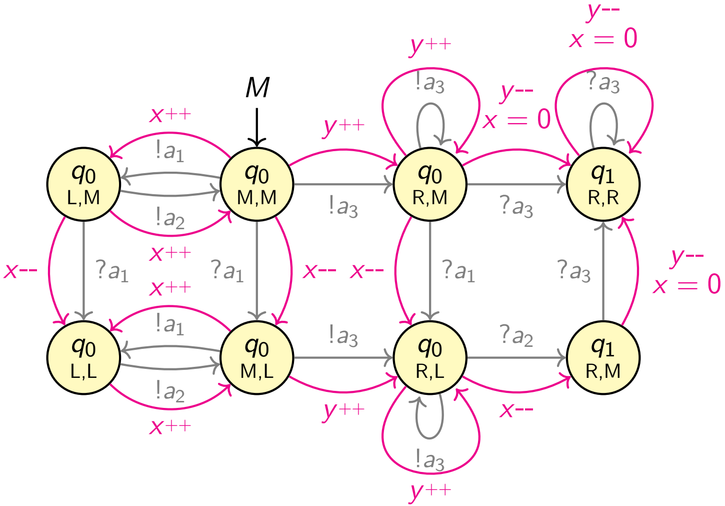


$$L = (a_1 a_2)^* a_3 a_3^*$$

Step 3: Normal form



Step 4: Conversion to counter machine



Equivalence between configurations

Given the normal form and counter automata, is there a 1-1 equivalence between the configurations?

Equivalence between configurations

Given the normal form and counter automata, is there a 1-1 equivalence between the configurations?
NO!
Given a counter configuration $(q; 3, 0)$ for some q, where $L = (ab)^*(c)^*$, what is the corresponding FIFO machine configuration?

Equivalence between configurations

Given the normal form and counter automata, is there a 1-1 equivalence between the configurations?

But we can keep track of the last message sent.

Equivalence between configurations

Given the normal form and counter automata, is there a 1-1 equivalence between the configurations?

$L_a^{\text{last}} \subseteq A_M^*$ be the set of words where a describes the **last sent** messages.

Equivalence between configurations

Given the normal form and counter automata, is there a 1-1 equivalence between the configurations?

$L_a^{\text{last}} \subseteq A_M^*$ be the set of words where a describes the **last sent** messages.

We can now conclude that runs in the FIFO machine are faithfully simulated by runs in the counter machine.

Other bounded problems

Table: Summary of key results. (D stands for Decidable)

	Letter-bounded	Bounded $\lvert Ch \rvert = 1$	Bounded $\lvert Ch \rvert > 1$
UNBOUND	D	D	D [5]
TERM	D	**EXPTIME**	**D**
REACH	D [6]	**EXPTIME**	**D, not ELEM**
CS-REACH	D	**EXPTIME**	D

[5] Jéron and Jard, *Testing for unboundedness of FIFO channels*, 1993.

[6] Gouda et al., *On deadlock detection in systems of communicating finite state machines*, 1987.

Future work

- Precise complexity for termination and boundedness
- Upper bounds for single channel case
- Output bounded reachability problems