Bounded Reachability Problems are Decidable in FIFO Machines

Benedikt Bollig   Alain Finkel   **Amrita Suresh**

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

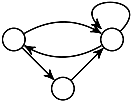
June 24, 2020

FIFO Machines

Distributed processes such that

- each process is a finite state machine

P_i :

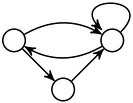


FIFO Machines

Distributed processes such that

- each process is a finite state machine
- there are a fixed number of processes

P_i :



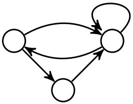
$P_1, \ldots, P_n$

FIFO Machines

Distributed processes such that

- each process is a finite state machine

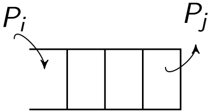
P_i :



- there are a fixed number of processes

$P_1, \ldots, P_n$

- they communicate using queues



FIFO Machines

- Studied since the 1980s. Widely used in distributed settings.

FIFO Machines

- Studied since the 1980s. Widely used in distributed settings.
- General model is difficult, hence underapproximations.

FIFO Machines

- Studied since the 1980s. Widely used in distributed settings.
- General model is difficult, hence underapproximations.
- Letter-bounded FIFO machines. [1]

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

FIFO Machines

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

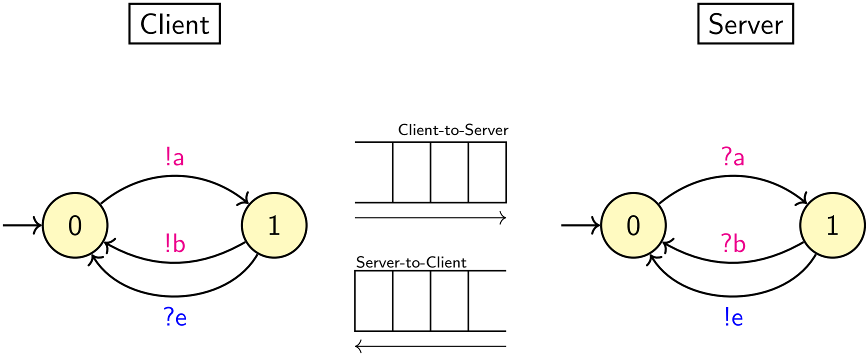
---

[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

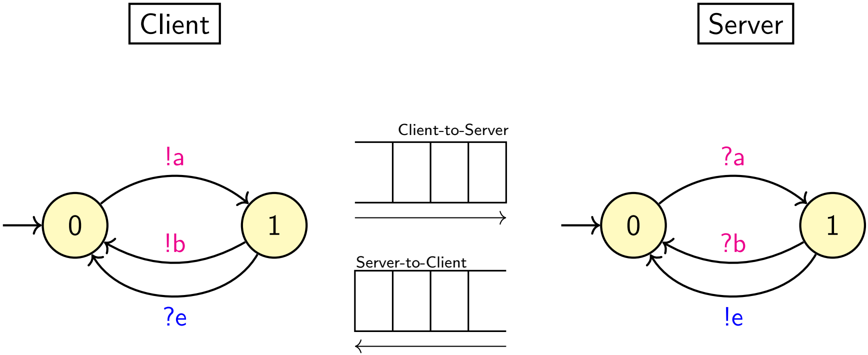
[3] Finkel and Praveen, *Verification of Flat FIFO Systems*, 2019

Example (Connection-Deconnection Protocol) [4]



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

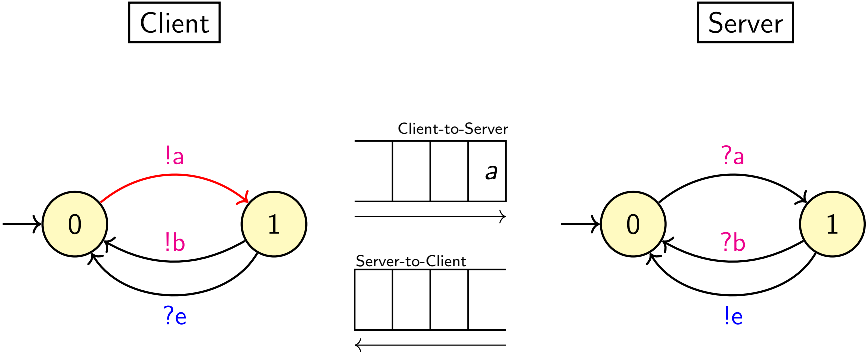
Example (Connection-Deconnection Protocol) [4]



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

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

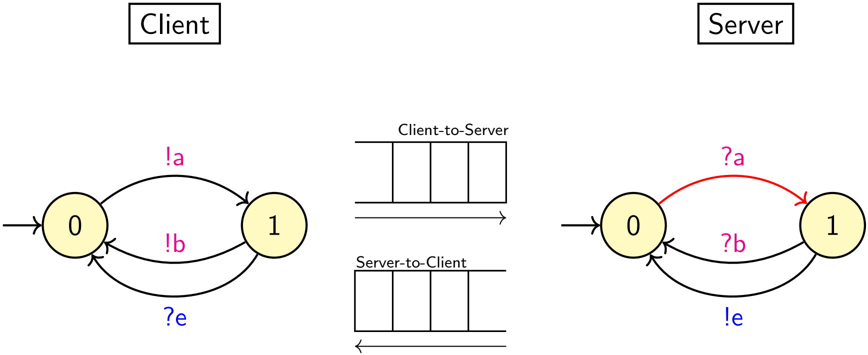
Example (Connection-Deconnection Protocol) [4]



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

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

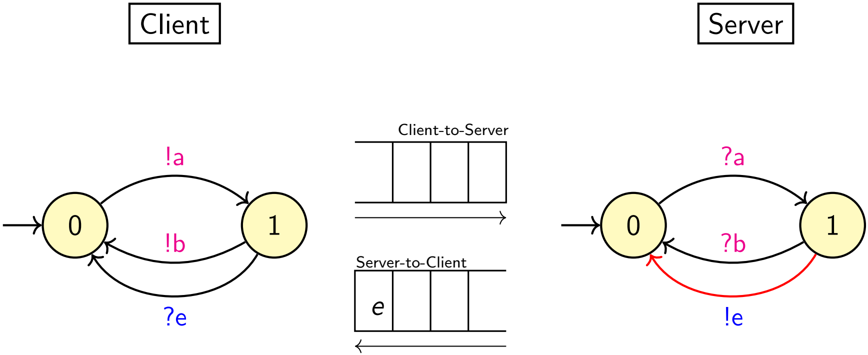
Example (Connection-Deconnection Protocol) [4]



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

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

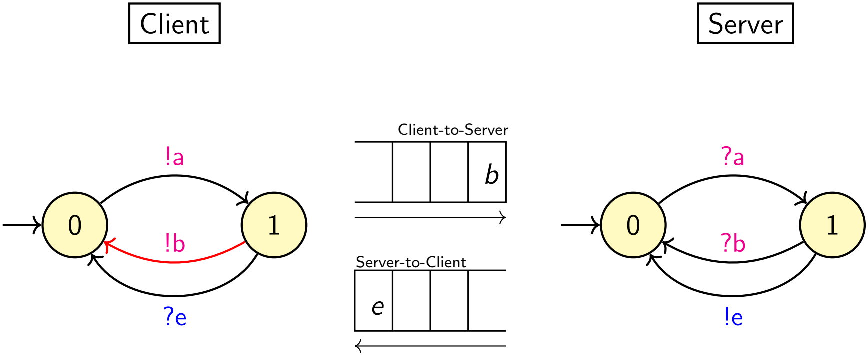
Example (Connection-Deconnection Protocol) [4]



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

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

Example (Connection-Deconnection Protocol) [4]

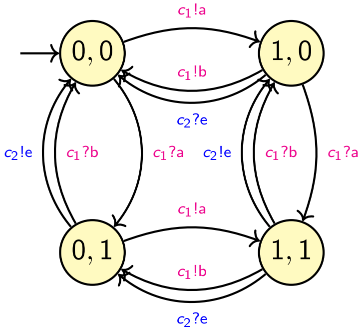


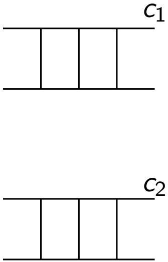
$$(0,0;\varepsilon,\varepsilon) \xrightarrow{!a} (1,0;a,\varepsilon) \xrightarrow{?a} (1,1;\varepsilon,\varepsilon) \xrightarrow{!e} (1,0;\varepsilon,e) \xrightarrow{!b} (0,0;b,e)$$

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

Formal model

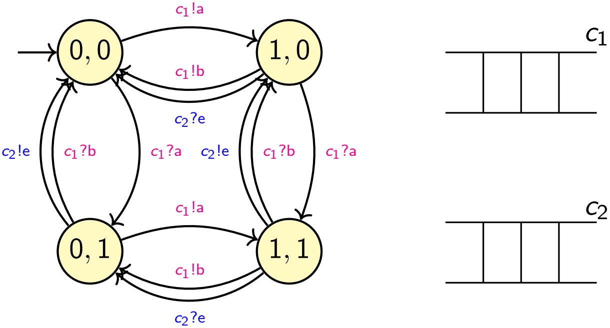
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where





Formal model

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

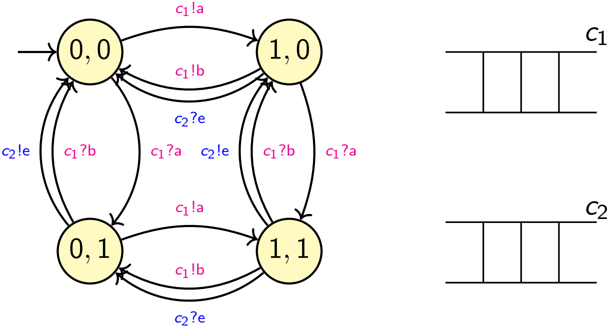


Q is a finite set of control-states.

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

Formal model

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

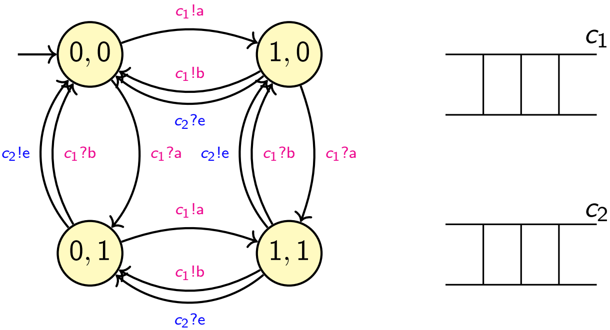


Ch is the number of channels.

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

Formal model

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

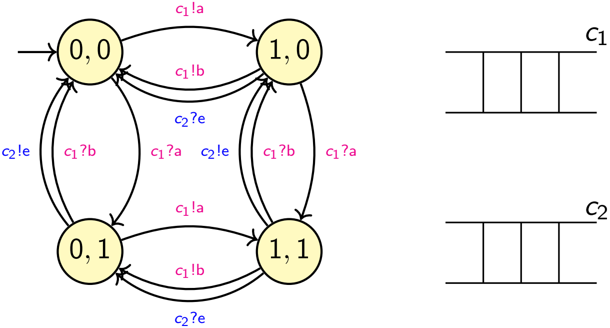


Σ is the alphabet.

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

Formal model

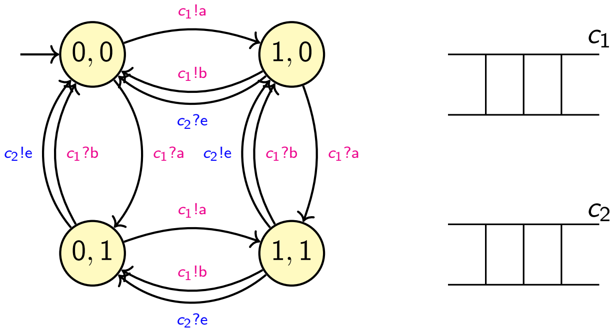
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where



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

Formal model

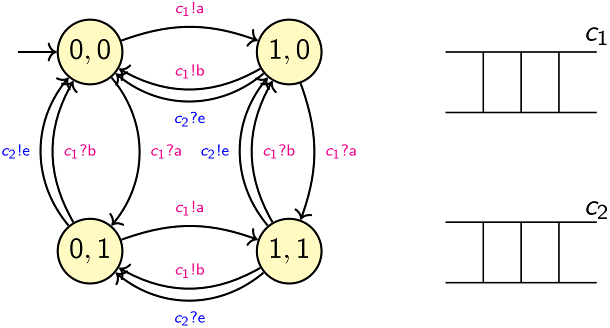
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where



$T \subseteq Q \times A_M \times Q$ 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 $(\mathbf{w_1}, ..., \mathbf{w_n})$ 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 $(\mathbf{w_1}, ..., \mathbf{w_n})$ 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 $(\mathbf{w_1}, ..., \mathbf{w_n})$ 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 $(\mathbf{w_1}, ..., \mathbf{w_n})$ 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 $(\mathbf{w_1}, ..., \mathbf{w_n})$ 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.)

Input-Bounded Reachability Problem

Given

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

Input-Bounded Reachability Problem

Given

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

Input-Bounded Reachability Problem

Given

- a FIFO machine $M = (Q, Ch, \Sigma, T, q_0)$,
- a control-state $q \in Q$,
- channel contents $\mathbf{w}$, and

Input-Bounded Reachability Problem

Given

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

Input-Bounded Reachability Problem

Given

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

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

Input-Bounded Reachability Problem

Theorem

The Input-Bounded Reachability Problem is decidable.

Input-Bounded Reachability Problem

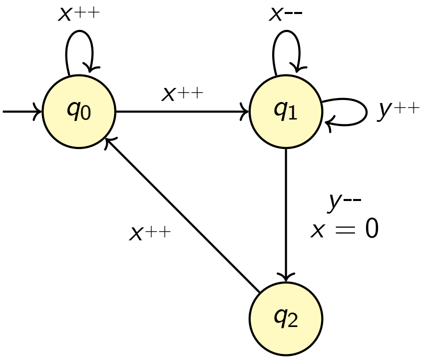
Theorem

The Input-Bounded Reachability Problem is decidable.

Proof using counter machines...

Counter machines

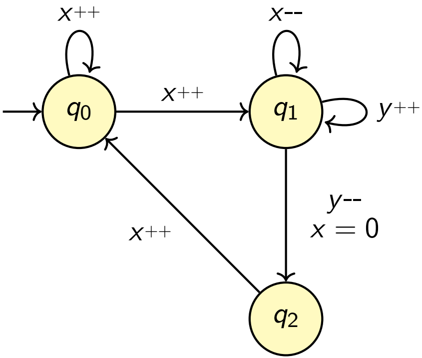
A counter machine (with zero tests) is a tuple $C = (Q, Cnt, T, q_0)$ where



- Q is the finite set of control-states. $Q = \{q_0, q_1, q_2\}$

Counter machines

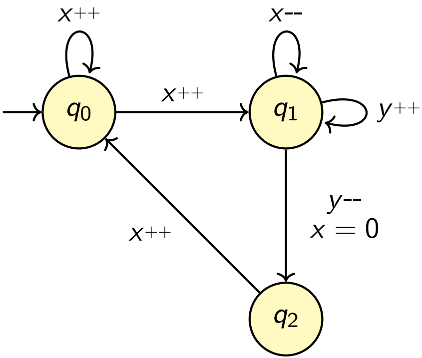
A counter machine (with zero tests) is a tuple $C = (Q, Cnt, T, q_0)$ where



- Cnt is a non-empty finite set of counters. $Cnt = \{x, y\}$

Counter machines

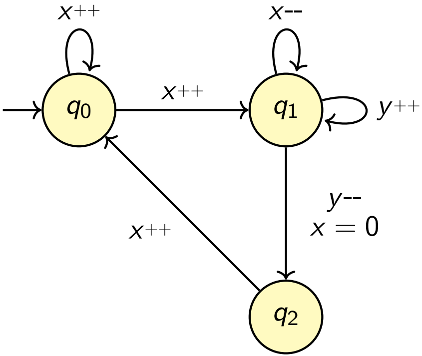
A counter machine (with zero tests) is a tuple $C = (Q, Cnt, T, q_0)$ where



- q_0 is the initial state.

Counter machines

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



- $T \subseteq Q \times A_C \times Q$ is the transition relation, where $A_C = \{x{+}{+}, x{-}{-} \mid x \in Cnt\} \times 2^{Cnt}$.

Counter machines with RESTRICTED zero tests

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

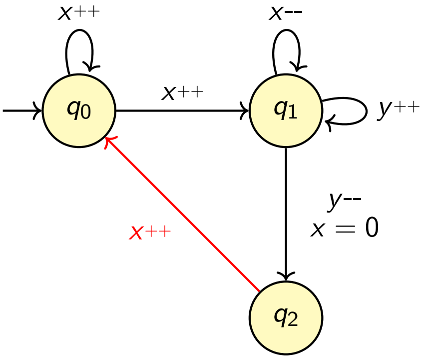
Counter machines with RESTRICTED zero tests

Formally, let L_{zero} be the set of words $(op_1(x_1), Z_1) \ldots (op_n(x_n), Z_n) \in A_C^*$.

Counter machines with RESTRICTED zero tests

Formally, let L_{zero} be the set of words $(op_1(x_1), Z_1) \dots (op_n(x_n), Z_n) \in A_C^*$. Then, for every two positions $1 \leq i \leq j \leq n$, we have $x_j \notin Z_i$.

Counter machines with RESTRICTED zero tests



Counter machines with RESTRICTED zero tests

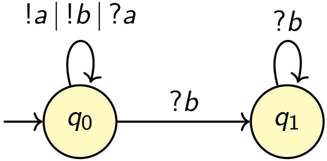
Theorem

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 $\mathbf{v}$, do we have $(q, \mathbf{v}) \in Reach_C(L_{zero} \cap L)$?

Translation

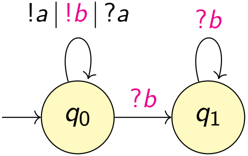
Intuition: Given a bounded language L, which is bounded over $(w_1, \ldots, w_n)$, we construct a counter x_i for each w_i.

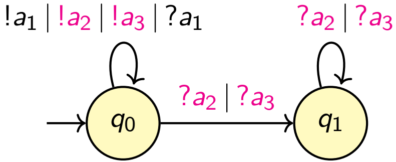
Translation



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

Step 1: Distinct letter property

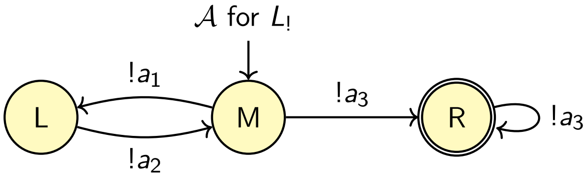




$\hat{L}_c = (ab)^* bb^*$

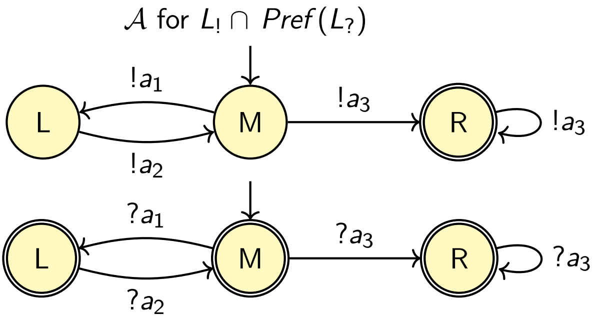
$L_c = (a_1 a_2)^* a_3 a_3{}^*$

Step 2: Trace property



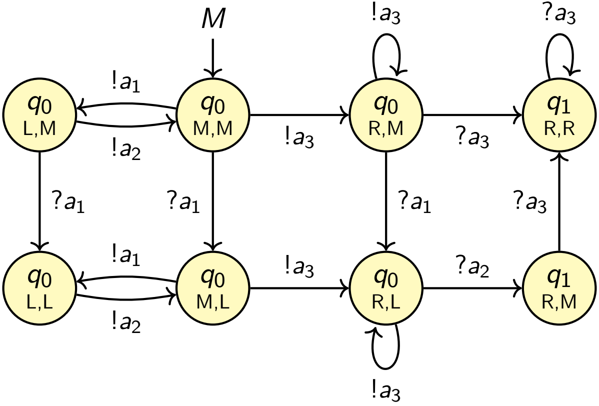
$$L_c = (a_1 a_2)^* a_3 a_3^*$$

Step 2: Trace property



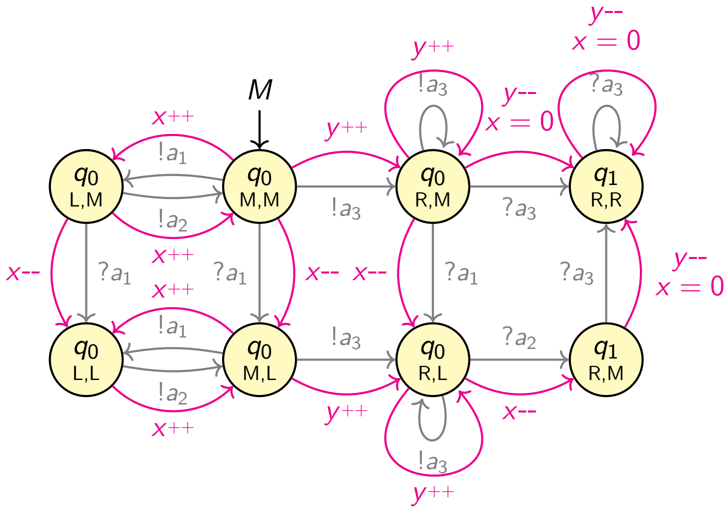
$$L_c = (a_1 a_2)^* a_3 a_3^*$$

Step 3: Normal form



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

Step 4: Conversion to counter machine



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

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.

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