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
FIFO Machines

Distributed processes such that

- Each process $P_i$ is a finite state machine
- There are a fixed number of processes $P_1, \ldots, P_n$
FIFO Machines

Distributed processes such that

- Each process $P_i$ is a finite state machine
- There are a fixed number of processes
- They communicate using FIFO queues
FIFO Machines

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

---

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

²Esparza et al., *Perfect Model for Bounded Verification*, 2012
³Finkel 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

---

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.

² Esparza et al., *Perfect Model for Bounded Verification*, 2012
³ Finkel and Praveen, *Verification of Flat FIFO Systems*, 2019
Example (Connection-Deconnection Protocol)  

\[\text{Client-to-Server} \]

\[\text{Server-to-Client} \]

\[!a \rightarrow ((1, 0), (a, \epsilon)) \]

\[?a \rightarrow ((1, 1), (\epsilon, \epsilon)) \]

\[!e \rightarrow ((1, 0), (\epsilon, e)) \]

\[!b \rightarrow ((0, 0), (b, e)) \]

---

\[\text{Jéron, Testing for unboundedness of FIFO channels, 1991.}\]
Example (Connection-Deconnection Protocol) \(^4\)

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

Example (Connection-Deconnection Protocol) \(^4\)

```
\(\begin{align*}
\text{Run} & \text{ - } (0, 0; \varepsilon, \varepsilon) \xrightarrow{!a} (1, 0; a, \varepsilon) \\
\end{align*}\)
```

Example (Connection-Deconnection Protocol) \(^4\)

\[(0, 0; \epsilon, \epsilon) \xrightarrow{!a} (1, 0; a, \epsilon) \xrightarrow{?a} (1, 1; \epsilon, \epsilon)\]

---

Example (Connection-Deconnection Protocol) \(^4\)

\[
\begin{align*}
(0, 0; \varepsilon, \varepsilon) & \xrightarrow{!a} (1, 0; a, \varepsilon) \xrightarrow{?a} (1, 1; \varepsilon, \varepsilon) \xrightarrow{!e} (1, 0; \varepsilon, e)
\end{align*}
\]

Example (Connection-Deconnection Protocol) \(^4\)

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

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

$$Q = \{(0, 0), (0, 1), (1, 0), (1, 1)\}.$$
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where $Ch$ is the number of channels.

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

$Ch$ is the number of channels.
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where

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

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

$\Sigma = \{a, b, e\}$. 
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
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where

\[
T \subseteq Q \times A_M \times Q \text{ is the transition relation where } \\
A_M = \{ c!a \mid a \in \Sigma \text{ and } c \in Ch \} \cup \{ c?a \mid a \in \Sigma \text{ and } c \in Ch \}
\]
A FIFO machine is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where

$q_0$ is the initial state.

$q_0 = (0, 0)$. 

\[ q_0 = (0, 0). \]
A configuration is \((q, w)\) where \(q\) is the control-state and \(w\) is a tuple of the channel contents. The set of configurations is \(S_M\).
Configurations and Reachability

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

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

\[ \text{Reach}_M = \{ s \in S_M \mid (q_0, \varepsilon) \overset{\sigma}{\rightarrow} s \text{ for some } \sigma \in A_M^* \} . \]

**Theorem**

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

Problem Definition

Configurations and Reachability

- $\text{Reach}_M(\sigma) = \{s \in S_M \mid (q_0, \varepsilon) \xrightarrow{\sigma} s\}$ where $\sigma \in A_M^*$. 
Problem Definition

Configurations and Reachability

- $\text{Reach}_M(\sigma) = \{ s \in S_M \mid (q_0, \varepsilon) \xrightarrow{\sigma} s \}$ where $\sigma \in A_M^*$.  
- $\text{Reach}_M(L) = \bigcup_{\sigma \in L} \text{Reach}_M(\sigma)$. 
Configurations and Reachability

We define the *send projection over c* \( \text{proj}_c! : \mathcal{A}_M^* \rightarrow \Sigma^* \)

Example: \( \text{proj}_c!(c! x. d! y. c? x. c! z. c! z) = xzz \)
Let $w_1, \ldots, w_n \in \Sigma^+$ be non-empty words where $n \geq 1$. 
$L$ is a **bounded language** over $(w_1, \ldots, w_n)$ if $L \subseteq w_1^* \ldots w_n^*$. 
Let \( w_1, \ldots, w_n \in \Sigma^+ \) be non-empty words where \( n \geq 1 \).

\( L \) is a **bounded language** over \((w_1, \ldots, w_n)\) if \( L \subseteq w_1^* \ldots w_n^* \).

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

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

$((ab)^*(cd)^*))^*$ is not a 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 $\Sigma$. 
$L! = \{w \in A_M^* | \text{proj}_c(w) \in L_c \text{ for all } c \in Ch\}$. 


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 $\Sigma$. $L! = \{ w \in A_M^* \mid proj_c!(w) \in L_c \text{ for all } c \in Ch \}$. (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 $\mathcal{R} \subseteq \prod_{c \in \text{Ch}} \Sigma_c^*$ and $\Theta = \prod_{c \in \text{Ch}} (\Sigma_c \cup \varepsilon)$.
We say that $\mathcal{R}$ is rational if there is a regular word language $R \in \Theta^*$ such that

$$\mathcal{R} = \{(a_1^1 \cdots a_n^1 \cdots a_1^n \cdots a_n^n)_{c \in \text{Ch}} \mid a_1^1 \cdots a_n^n \in R \text{ with } n \in \mathbb{N} \text{ and } a_i^j = (a_i^j)_c \subseteq \text{Ch} \in \Theta \text{ for } i \in \{1, \ldots, n\}\}.$$
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

$$\mathcal{R} = \{(a_1^c \ldots a_n^c)_{c \in Ch} \mid a_1^1 \ldots a_n^n \in R \text{ with } n \in \mathbb{N} \text{ and } a_i^i = (a_i^c)_{c \in Ch} \in \Theta \text{ for } i \in \{1, \ldots, n\}\}.$$ 

Example: $\mathcal{R} = \{(a^m, b^n \mid m \geq n)\}$ is a rational relation, witnessed by $R = ((a, b) + (a, \varepsilon))^*$. 
Input-Bounded Rational Reachability Problem

Given

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

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 $\Sigma$,
- a rational relation $R \subseteq \prod_{c \in Ch} \Sigma^*$. 

Amrita Suresh
Bounded Reachability Problems in FIFO Machines 8 / 18
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 $\Sigma$, 
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 $\Sigma$,
- a rational relation $\mathcal{R} \subseteq \prod_{c \in Ch} \Sigma^*$. 

Amrita Suresh
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 $\Sigma$,
- a rational relation $\mathcal{R} \subseteq \prod_{c \in Ch} \Sigma_c^*$.

Question: Do we have $(q, w) \in \text{Reach}_M(L!)$ for some $w \in \mathcal{R}$?
Problem Definition

Input-Bounded Rational Reachability Problem

Theorem

The Input-Bounded Rational Reachability Problem is decidable.
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

- Transition from $q_0$ to $q_1$: $x++$
- Transition from $q_1$ to $q_0$: $x--$
- Transition from $q_1$ to $q_2$: $y+$
- Transition from $q_2$ to $q_0$: $x = 0$

- $q_0$ to $q_1$: $x++$
- $q_1$ to $q_2$: $y--$

Amrita Suresh
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 $(q, v) \in \text{Reach}_C(L_{\text{zero}} \cap L)$?
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

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

\[ L = (a_1a_2)^* a_3a_3^* \]
Step 2: Trace property

$L = (a_1 a_2)^* a_3 a_3^*$
Step 2: Trace property

\[ L = (a_1 a_2)^* a_3 a_3^* \]
Step 3: Normal form

Reduction to Counters

Amrita Suresh

Bounded Reachability Problems in FIFO Machines

14 / 18
Step 4: Conversion to counter machine

Reduction to Counters
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?

$L_{\text{last}}^{a} \subseteq A_{M}^{*}$ 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 $|Ch| = 1$ | Bounded $|Ch| > 1$ |
|------------------|----------------|------------|----------------|
| UNBOUND          | D              | D          | $D^5$          |
| TERM             | D              | $\text{EXPTIME}$ | $D$          |
| REACH            | $D^6$          | $\text{EXPTIME}$ | $D$, not $\text{ELEM}$ |
| CS-REACH         | D              | $\text{EXPTIME}$ | $D$          |

---


Future work

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