Bounded Reachability Problems are Decidable in FIFO Machines

Amrita Suresh
Université Paris-Saclay, ENS Paris-Saclay, CNRS
Laboratoire Spécification et Vérification, 91190, Gif-sur-Yvette, France
amrita.suresh@ens-paris-saclay.fr

Acknowledgements.

This work was done in collaboration with Benedikt Bollig and Alain Finkel, at the LSV, CNRS, ENS Paris-Saclay.

1 Introduction

Asynchronous distributed processes communicating using First In First Out (FIFO) channels are being widely used for distributed and concurrent programming. Since such systems of communicating processes, which communicate through (at least two) one-directional FIFO channels, can simulate Turing machines, most verification properties, such as testing the unboundedness of a channel, are undecidable for them [2, 12, 13].

Many papers from the 1980s to today have studied FIFO systems in which the input-language of a channel (i.e. the set of words that enter in a channel) is included in the set of prefixes $Pref(B)$ of a particular bounded language $B = w_1^* w_2^* ... w_n^*$. We call this class of FIFO machines *input-bounded*. When the *set of letters* that may enter in a channel c is reduced to a unique letter a_c, then the input-language of c is included in a_c^* and this subclass trivially reduces to VASS (Vector Addition Systems with States) and Petri nets [14]. A variant of the reachability problem, the deadlock problem, is shown decidable for input-*letter*-bounded FIFO systems in [8]. There are some other subclasses of this model for which some classical properties were shown decidable, such as monogeneous FIFO nets [5], linear FIFO nets [6], and flat systems [4, 7].

We may use the previous decidability results as an underapproximation for any general FIFO machines over bounded languages. While all the executions of the machine may not be input-bounded, we can use our methods to verify whether the executions conforming to this condition satisfy a given property. Moreover, if there is a bug in the restricted reachability set (or an unfavourable configuration is reached via an input-bounded execution), we can immediately deduce that the original machine is unsafe.

Our contributions: We solve a problem left open in [8] regarding the decidability of the reachability problem for input-bounded FIFO machines. We construct a simulation of input-bounded FIFO machines by counter machines with restricted zero tests. We extend this result to other verification properties like unboundedness, control-state reachability and termination.

2 Bounded reachability

We consider FIFO machines having a sequential control-graph rather than distributed systems of communicating processes. It is clear that given a distributed system, one may compute the Cartesian product of all processes and obtain a FIFO machine (the converse is not always true).

▶ **Definition 1.** *A* FIFO machine *is a tuple $M = (Q, Ch, \Sigma, T, q_0)$ where Q is the finite set of* control-states*, $q_0 \in Q$ is the* initial state*, Ch is the non-empty finite set of channels, Σ is the* message alphabet, *and* $T \subseteq Q \times A_M \times Q$ *is the transition relation where* $A_M = \{\langle c!a\rangle \mid a \in \Sigma$ *and* $c \in Ch\} \cup \{\langle c?a\rangle \mid a \in \Sigma$ *and* $c \in Ch\}$ *is the set of send and receive actions that can be executed in* M*.*

▶ **Example 2** (Connection Deconnection Protocol)**.** The (simplified) connection-deconnection protocol, CDP, between two machines is described as follows (see Fig. 1): The first machine (on the left) can send the message "a" (denotes opening a session) to the other machine. If the first machine closes its own session, it sends the message "b" to the other machine. The second machine can read these messages, and ask for session closure by sending "e". This protocol has been studied in [9].

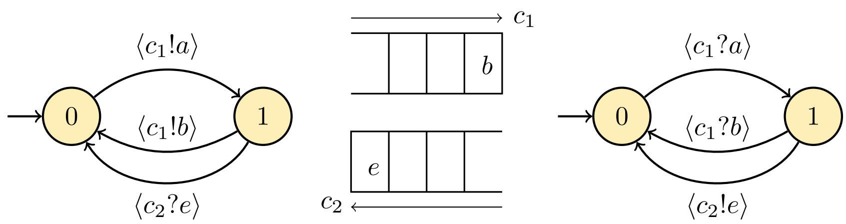

Figure 1 The connection-deconnection protocol

The FIFO machine M induces a (potentially infinite) transition system. Its set of configurations is $S_M = Q \times (\Sigma^*)^{Ch}$. In $(q, \mathbf{w}) \in S_M$, the first component q denotes the current control-state and $\mathbf{w} = (\mathbf{w}_c)_{c\in Ch}$ determines the contents $\mathbf{w}_c$ for every channel $c \in Ch$. The initial configuration is $init_M = (q_0, \boldsymbol{\varepsilon})$ where $\boldsymbol{\varepsilon} = (\varepsilon, \ldots, \varepsilon)$, i.e. every channel is empty.

An example run of the machine M_1 (which is the Cartesian product of the two automata) in Example 2 is as follows. We represent a configuration as $(q, \mathbf{w})$ where q is a tuple of the states of the automata and $\mathbf{w}$ is a tuple of the channel contents. Thus, the initial configuration is $((0, 0), (\varepsilon, \varepsilon))$. Let σ be a finitely long run as follows: $((0, 0), (\varepsilon, \varepsilon)) \xrightarrow{\langle c_1!a\rangle} ((1, 0), (a, \varepsilon)) \xrightarrow{\langle c_1?a\rangle} ((1, 1), (\varepsilon, \varepsilon)) \xrightarrow{\langle c_2!e\rangle} ((1, 0), (\varepsilon, e))$. We can express it as $((0, 0), (\varepsilon, \varepsilon)) \xrightarrow{\sigma} ((1, 0), (\varepsilon, e))$.

We let $Reach_M(w) = \{s \in S_M \mid (q_0, \boldsymbol{\varepsilon}) \xrightarrow{w} s\}$ for $w \in A_M^*$, and for $L \subseteq A_M^*$, we have $Reach_M(L) = \bigcup_{w\in L} Reach_M(w)$.

▶ **Definition 3.** *Let* $w_1, \ldots, w_n \in \Sigma^+$ *be non-empty words where* $n \geq 1$*. A* bounded language *over* $(w_1, \ldots, w_n)$ *is a language* $L \subseteq w_1^* \ldots w_n^*$*.*

Consider a FIFO machine $M = (Q, Ch, \Sigma, T, q_0)$. For $c \in Ch$, we let $proj_{c!} : A_M^* \to \Sigma^*$ be the morphism defined by $proj_{c!}(\beta) = a$ if $\beta = \langle c!a\rangle \in A_M$, and $proj_{c!}(\beta) = \varepsilon$ if $\beta \in A_M$ is not of the form $\langle c!a\rangle$ for some $a \in \Sigma$. We define $proj_{c?}(\sigma) \in \Sigma^*$ accordingly. With this, given a tuple $\mathscr{L} = (L_c)_{c\in Ch}$ of bounded languages $L_c \subseteq \Sigma^*$, we set $\mathscr{L}_! = \{w \in A_M^* \mid proj_{c!}(w) \in L_c$ for all $c \in Ch\}$ and $\mathscr{L}_? = \{w \in A_M^* \mid proj_{c?}(w) \in L_c$ for all $c \in Ch\}$. We observe that, if all L_c are regular, then so are $\mathscr{L}_!$ and $\mathscr{L}_?$.

Our reachability problem asks whether a given configuration $(q, \mathbf{w})$ is reachable along a sequence of actions from $\mathscr{L}_!$, i.e., whether $(q, \mathbf{w}) \in Reach_M(\mathscr{L}_!)$. Note that, if $(q_0, \boldsymbol{\varepsilon}) \xrightarrow{\sigma}_M (q', \mathbf{w}')$ and $\sigma \in \mathscr{L}_!$, then we also have $\sigma \in Pref(\mathscr{L}_?)$. Thus, $Reach_M(\mathscr{L}_!) = Reach_M(\mathscr{L}_! \cap Pref(\mathscr{L}_?))$ so that we can restrict to action sequences from $\mathscr{L}_! \cap Pref(\mathscr{L}_?)$.

▶ **Definition 4** (Input-Bounded (IB) reachability problem)**.** *The* IB reachability problem *is defined as follows: Given a FIFO machine* $M = (Q, Ch, \Sigma, T, q_0)$*, a control-state* $q \in Q$*, channel contents* $\mathbf{w}$*, and a tuple* $\mathscr{L} = (L_c)_{c\in Ch}$ *of non-empty regular bounded languages over* Σ*, do we have* $(q, \mathbf{w}) \in Reach_M(\mathscr{L}_!)$*?*

We prove that IB reachability is indeed decidable. The overarching idea of the proof is to construct a counter machine which models the FIFO machine. Since we are considering only executions that belong to a tuple of bounded languages, the set of words that enter a channel are included in $w_1^* \dots w_n^*$ for some words $w_1, \dots, w_n \in \Sigma^*$. In the corresponding machine, we have a counter for each word $w_1, \dots, w_n$. These counters are incremented every time a letter associated to these words is sent to the channel, and decremented if the letter is received from the channel. Furthermore, we need to ensure the FIFO property of the channel, i.e. a letter from w_i is received only if no letters from words $w_1, \dots, w_{i-1}$ are present in the channel. This is done by adding zero tests for the counters. Since the language is bounded, we show that we can impose a restriction on these zero test. Thus, the question of reachability of a configuration $(q, \mathbf{w})$ now corresponds to the reachability of a configuration in the associated counter machine (with restricted zero tests).

Reachability in presence of these restricted zero tests straightforwardly reduces to configuration-reachability in classical counter machines without zero tests (i.e., VASS and Petri nets) by delaying the zero tests to the end of the run and checking only once. The latter is known to be decidable [11], though inherently non-elementary [3]. Hence, we can deduce the following result.

▶ **Theorem 5.** *IB reachability is decidable for FIFO machines.*

We then extend this result to prove the decidability of the control-state reachability problem. We also prove the decidability of boundedness and termination problems by reducing them to reachability. Furthermore, we show that for single channels, the decidability of most of these problems is in EXPTIME (although it remains to be seen what the tight upper bound is). This is shown by reducing them to a special case of Unary Ordered Multi Pushdown Systems (UOMPDS) [1]. And even in the single channel case, all the problems are NP hard. Our results are tabulated below. (By D, we mean the result is decidable, but we do not know precisely the complexity of the problem.)

Table 1 Summary of key results; results for all other extensions are subsumed by these results.

	Letter-bounded	Flat	Bounded $\lvert Ch\rvert = 1$	Bounded $\lvert Ch\rvert > 1$
UNBOUND	D [8]	NP-C [7]	D	D[10]
TERM	D	NP-C [7]	**EXPTIME**	**D**
REACH	D	NP-C [7]	**EXPTIME**	**D, not ELEM**
CS-REACH	D	NP-C [4, 7]	**EXPTIME**	D

3 Conclusion and perspectives

We extend recent results of the *bounded verification* of FIFO machines [4] and of *flat* FIFO machines [7] by using bounded languages for controlling the input-languages of FIFO channels (and not for controlling the runs of the machine). The approach of understanding FIFO systems using underapproximations is an interesting way forward. We have a model which subsumes a lot of existing subclasses, with decidable properties, and verification of general FIFO systems may be possible by the analysis of these approximations using a semi-decision algorithm. Another approach forward would be to analyse the complexity of these problems, such as finding a tighter upper bound for the reachability in the single channel case, and checking if the complexity of unboundedness for the multiple channels case is in EXPSPACE, as for VASS.

References

1. Mohamed Faouzi Atig, K. Narayan Kumar, and Prakash Saivasan. Adjacent ordered multi-pushdown systems. *Int. J. Found. Comput. Sci.*, 25(8):1083–1096, 2014. URL: `https://doi.org/10.1142/S0129054114400255`, `doi:10.1142/S0129054114400255`.
2. Daniel Brand and Pitro Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983. URL: `https://doi.org/10.1145/322374.322380`, `doi:10.1145/322374.322380`.
3. Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for Petri nets is not elementary. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 24–33. ACM, 2019. URL: `https://doi.org/10.1145/3313276.3316369`, `doi:10.1145/3313276.3316369`.
4. Javier Esparza, Pierre Ganty, and Rupak Majumdar. A perfect model for bounded verification. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 285–294. IEEE Computer Society, 2012. URL: `https://doi.org/10.1109/LICS.2012.39`, `doi:10.1109/LICS.2012.39`.
5. Alain Finkel. About monogeneous FIFO Petri nets. In *Proceedings of the 3rd International Conference on Applications and Theory of Petri Nets (APN'82)*, Varenna, Italy, September 1982.
6. Alain Finkel and Annie Choquet. Simulation of linear FIFO nets by Petri nets having a structured set of terminal markings. In *Proceedings of the 8th International Conference on Applications and Theory of Petri Nets (APN'87)*, Zaragoza, Spain, June 1987.
7. Alain Finkel and M. Praveen. Verification of flat FIFO systems. In Wan Fokkink and Rob van Glabbeek, editors, *30th International Conference on Concurrency Theory, CONCUR 2019, August 27-30, 2019, Amsterdam, the Netherlands*, volume 140 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: `https://doi.org/10.4230/LIPIcs.CONCUR.2019.12`, `doi:10.4230/LIPIcs.CONCUR.2019.12`.
8. M. G. Gouda, E. M. Gurari, T. H. Lai, and L. E. Rosier. On deadlock detection in systems of communicating finite state machines. *Comput. Artif. Intell.*, 6(3):209–228, July 1987.
9. Thierry Jéron. Testing for unboundedness of FIFO channels. In Christian Choffrut and Matthias Jantzen, editors, *STACS 91, 8th Annual Symposium on Theoretical Aspects of Computer Science, Hamburg, Germany, February 14-16, 1991, Proceedings*, volume 480 of *Lecture Notes in Computer Science*, pages 322–333. Springer, 1991. URL: `https://doi.org/10.1007/BFb0020809`, `doi:10.1007/BFb0020809`.
10. Thierry Jéron and Claude Jard. Testing for unboundedness of FIFO channels. *Theor. Comput. Sci.*, 113(1):93–117, 1993. URL: `https://doi.org/10.1016/0304-3975(93)90212-C`, `doi:10.1016/0304-3975(93)90212-C`.
11. Ernst W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984.
12. Gérard Memmi and Alain Finkel. An introduction to FIFO nets and monogeneous nets: A subclass of FIFO nets. *Theor. Comput. Sci.*, 35:191–214, 1985. URL: `https://doi.org/10.1016/0304-3975(85)90014-3`, `doi:10.1016/0304-3975(85)90014-3`.
13. Bernard Vauquelin and Paul Franchi-Zannettacci. Automates a file. *Theor. Comput. Sci.*, 11:221–225, 1980. URL: `https://doi.org/10.1016/0304-3975(80)90047-X`, `doi:10.1016/0304-3975(80)90047-X`.
14. Yao-Tin Yu and Mohamed G. Gouda. Unboundedness detection for a class of communicating finite-state machines. *Inf. Process. Lett.*, 17(5):235–240, 1983. URL: `https://doi.org/10.1016/0020-0190(83)90105-9`, `doi:10.1016/0020-0190(83)90105-9`.