Verification of FIFO Systems

Amrita Suresh[1]
Supervised by Benedikt Bollig[1] and Alain Finkel [1,2]

[1] Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMF, France
[2] Institut Universitaire de France

12 May 2021

Abstract

Asynchronous distributed processes communicating using First In First Out (FIFO) channels are widely used for distributed and concurrent programming. However, since they can simulate Turing machines, most verification properties are undecidable for them. Hence, there is a need to analyze subclasses which are powerful yet have decidable properties. Towards this goal, we analyze two underapproximations of general FIFO systems: 1) Systems with input-bounded runs (i.e. the sequence of messages sent through a particular channel belongs to a given bounded language), and 2) Synchronizable systems (i.e. if every execution can be rescheduled so that it meets certain criteria, e.g., a channel bound). For the former, we have shown that rational-reachability (and by extension, control-state reachability, deadlock, boundedness, etc.) are decidable. For the latter, we provide a framework, that unifies existing definitions, and allows one to easily derive decidability results for synchronizability. We plan to finish the study on these systems, and then analyze the same problems for reversal-bounded systems along with its complexity in order to construct a tool.

1 Introduction

Communication with asynchronous message passing is widely used in concurrent and distributed programs implementing various types of systems such as communication protocols [41], hardware design, MPI programs, and more recently for designing and verifying session types [34], web contracts, choreographies, concurrent programs, Erlang, Rust, etc. An asynchronous message passing system is built as a set of processes running in parallel, communicating asynchronously by sending messages to each other via channels or message buffers. Messages sent to a given process are stored in its entry buffer, waiting for the moment they will be received by the process. In general, sending messages is not blocking for the sender process, which means that the message buffers are supposed to be of unbounded size. It is well-known that such programs are hard to get right. Indeed, asynchrony introduces a tremendous amount of new possible interleavings between actions of parallel processes, and makes very hard to analyze the effect of all their computations. In particular, when buffer are ordered (FIFO buffers), the verification of reachability queries is undecidable even when each of the processes is finite-state [9]. This gives rise to the interest of studying underapproximations, which restrict the behavior of general FIFO systems in turn having decidable verification properties.

1.1 Communicating Systems and Message Sequence Charts

We define a message passing system as the composition of a set of processes that exchange messages, which can be stored in FIFO buffers before being received. Each process is described as a state machine that evolves by executing send or receive actions. An execution of such a system can be represented abstractly using a partially-ordered set of events, called a trace. The partial order in a trace represents the causal relation between events. We show that these systems satisfy causal delivery, i.e., the order in which messages are received by a process is consistent with the causal relation between the corresponding sendings.

Message Sequence Charts (or MSCs) provide a trace language for the specification and description of the communication behavior of system components and their environment by means of message interchange. Processes are represented as vertical lines, and horizontal arrows (which we call events) represent messages from one component to another.

1.2 Reachability in FIFO systems

If one restricts to runs with B-bounded channels (the number of messages in every channel does not exceed B), then reachability becomes decidable for existentially-bounded and universally-bounded FIFO systems [23]. When limiting the number of phases, the bounded-context reachability problem is in 2-EXPTIME, even for recursive FIFO systems [33, 29]. For non-confluent topology, reachability is in EXPTIME for recursive FIFO systems with 1-bounded channels [29]. The notion of k-synchronous computations was introduced in [8]. Reachability under this restriction and checking k-synchronizability are both PSPACE-complete [26]. Reachability is in PTIME in half-duplex systems [10] with two processes (moreover, the reachability set is recognizable and effectively computable), but the natural extension to three processes leads to undecidability. Lossy FIFO systems (where the channels can lose messages) [1, 18] have been shown to be well-structured and have a decidable (but non-elementary) reachability problem [11]. In [37, 2], uniform criteria for decidability of reachability and model-checking questions are established for communicating recursive systems whose restricted architecture or communication mechanism gives rise to behaviors of bounded tree-width.

1.3 MSO Logic and Special Tree-width

In mathematical logic, monadic second-order logic (MSO) is the fragment of second-order logic where the second-order quantification is limited to quantification over sets.

The main appeal of the study of this language is that it is decidable for many sets of (finite or infinite) structures. It is particularly important in the logic of graphs, because of Courcelle's theorem [12], which provides algorithms for evaluating monadic second-order formulas over graphs of bounded tree-width.

Second-order logic allows quantification over predicates. However, MSO is the fragment in which second-order quantification is limited to monadic predicates (predicates having a single argument). This is often described as quantification over "sets" because monadic predicates are equivalent in expressive power to sets (the set of elements for which the predicate is true).

Special tree-width [13], is a graph measure that indicates how close a graph is to a tree (we may also use classical *tree-width* instead). This or similar measures are commonly employed in verification. For instance, tree-width and split-width have been used in [14] to reason about graph behaviors generated by pushdown systems. Here we wish to apply it to reason about MSCs. We adopt the following game based definition from [7].

Adam and Eve play a two-player turn based "decomposition game" whose positions are MSCs with some pebbles placed on some events. Eve's positions are *marked MSC fragments* (an MSC with possibly some edges removed) and U the subset of marked events. Adam's positions are pairs of marked MSC fragments. Eve can choose to mark some events of the MSC, remove some edges, and/or divide the MSC fragment into two such that the original fragment is the disjoint union of the two resulting fragments. When it is Adam's turn, he simply chooses one of the two marked MSC fragments. We say that the game is k-winning for Eve if she has a (positional) strategy that allows her to reach a terminal position such that, in every single position visited along the play, including the final one, there are at most $k+1$ marked events.

2 Input-bounded systems

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 [9, 39, 40].

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 [42]. A variant of the reachability problem, the deadlock problem, is shown decidable for input-*letter*-bounded FIFO systems in [27]. There are some other subclasses of this model for which some classical properties were shown decidable, such as monogeneous FIFO nets [17], linear FIFO nets [19], and flat systems [16, 21].

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 unfavorable 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 [27] regarding the decidability of the reachability problem for input-bounded FIFO machines. It is formalized in the theorem below.

Theorem. *Input bounded rational-reachability is decidable for FIFO machines.*

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 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, 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 [38], though inherently non-elementary [15].

We extend this result to other verification properties like unboundedness, control-state reachability and termination. Moreover, we also extend reachability to rational reachability that allows to deduce the decidability of and deadlock. The results are summarized in the table below.

Table 1: Summary of key results; results for all other extensions are subsumed by these results (D stands for decidable).

	Flat	Letter-bounded	Bounded
UNBOUND	NP-C ([21])	D ([27])	D ([31])
TERM	NP-C ([21])	D	**D**
REACH	NP-C ([21])	D	**D, not ELEM**
CS-REACH	NP-C ([16, 21])	D	D
DEADLOCK	D	D ([27])	**D**

Furthermore, we study the natural dual of the input-bounded reachability problem, which are systems of output-bounded languages in which the set of words received by each channel is constrained to be bounded, and are able to deduce the reachability, unboundedness, termination and control-state reachability for the same.

We obtain better upper bounds for the input-bounded reachability of FIFO machines with a single channel (reachability is still undecidable for FIFO machines with a single channel). This is done by reducing it to reachability in *unary ordered multi-pushdown systems* (a class that was previously analyzed in [3]). It is, hence, solvable in EXPTIME.

Finally, following the bounded verification paradigm, applied to FIFO machines (for instance in [16, 21]), we open the way to a methodology that would apply existing results on input-bounded FIFO machines to general FIFO machines.

3 Synchronizable systems

In 2004, Lohrey and Muscholl introduced *existentially k-bounded* systems [36] (see also [35, 25, 24]) where each of its accepting executions leading to a stable (with empty channels) final configuration can be re-ordered into a k-bounded execution. This property is undecidable, even for a given k [24]. A more general definition, still called existentially bounded, is given in 2014 where the considered executions are *not* supposed to be final or stable [32]. In [36, 28], the notion of *universally k-bounded* (all accepting executions are k-bounded) is also discussed and the authors show that the property is undecidable in general. In 2011, Basu and Bultan introduced *synchronizable* systems [4], for which every execution is equivalent (for the projection on sending messages) to one of the same system but communicating by rendezvous; to avoid ambiguity, we call such systems *send-synchronizable.* In 2018, Bouajjani et al. said that a system $\mathcal{S}$ is *k-synchronizable* [8] (to avoid confusion we call such systems *weakly k-synchronizable*) if every MSC of $\mathcal{S}$ admits a linearization (which is not necessarily an execution) that can be divided into sections of at most k messages. After each section, a message is either read, or will never be read. This constraint seems to imply that buffers are bounded to k messages. However, as the linearization need not be an execution, it results that a weakly k-synchronizable execution, even with the more efficient reschedule, can need unbounded channels to be run by the system.

All these notions amount to asking whether all behaviors generated by a given communicating system have a particular shape, i.e., whether they are all included in a fixed (or given) set of MSCs. Thus, the synchronizability problem is essentially an inclusion problem.

Our contributions: We show that, for decidability, it is enough to have that the set of MSCs is MSO-definable and special-tree-width-bounded (STW-bounded). The main theorem can be summarized as follows:

Theorem. *Fix finite sets of processes $\mathbb{P}$ and messages $\mathbb{M}$. Let $\mathcal{C}$ be a MSO-definable and special-tree-width-bounded class (over $\mathbb{P}$ and $\mathbb{M}$). The following problem is decidable: Given a communicating system $\mathcal{S}$, and $L(\mathcal{S})$ be the language of $\mathcal{S}$ (defined directly as a set of MSCs), do we have $L(\mathcal{S}) \subseteq \mathcal{C}$?*

This general framework based on monadic second-order (MSO) logic and (special) tree-width captures most existing definitions of systems that may work with bounded channels and allows us to unify the notions of synchronizability. The framework allows to simplify the proofs and sometimes lets us extend the statement. Moreover, reachability and model checking are decidable in this framework.

We generalize the existing notion of (weak) k-synchronizability in [8] and we introduce three new classes of synchronizable systems, called weakly synchronizable (that are more general than weakly k-synchronizable), strongly synchronizable and strongly k-synchronizable (which are particular cases of weakly synchronizable). We then prove that these four properties all fit in our framework and then they are all decidable by using the same generic proof.

We then deduce that reachability and model checking are decidable for the six classes of systems (only the control-state was shown decidable for weakly k-synchronizable in [8] and it is clearly also decidable for existentially/universally bounded systems but reachability properties are generally not studied for these classes of systems).

A key difference between the works in literature is that they consider different communication architectures. Existentially bounded systems have been studied for p2p (with one queue per pair of processes), whereas k-synchronizability has been studied for mailbox communication, for which each process merges all its incoming messages in a unique queue. Moreover, variants of those definitions can be obtained depending on if we consider messages that are sent but never read, called unmatched messages. We provide a comparison between the six synchronizable classes both for p2p and mailbox semantics (as illustrated below). In particular, we clarify the link between weakly synchronizable and existentially bounded systems for both p2p and mailbox systems, which has been left open in literature.

Figure 1: Hierarchy of classes for p2p systems

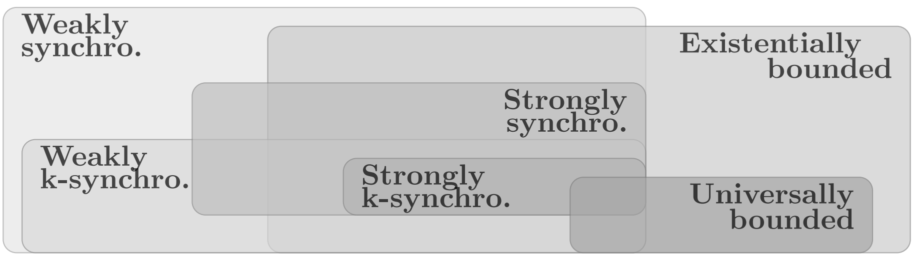

Figure 2: Hierarchy of classes for mailbox systems

Finally, in order to obtain better complexity results for some classes (strongly and weakly synchronizable systems), we also use the fragment of propositional dynamic logic with loop and converse (LCPDL) instead of MSO logic in our framework.

4 Future work

Our research plans for the second half of this PhD is as follows (including a rough time line):

- **June 2021 - August 2021:** Complete the study on the decidability of synchronizability and branch WSTS. We plan to submit two papers by August to an international conference/journal. This effort has already commenced, when we began the literature survey on synchronizability and the initial work on input-bounded systems. We are looking to explore the decidability of reachability in send-synchronizable systems as described in [5, 20].
- **August 2021 - December 2021:** Study a third subclass of FIFO machines known as reversal bounded FIFO systems. This class was introduced in [30], and more recently in [22, 6]. It seems like reversal boundedness could be a more efficient underapproximation for FIFO systems. If this analysis proves to be more succinct, we could construct a based on flat, input-bounded and reversal-bounded FIFO systems with use of existing softwares and algorithms. There is a lack of verification tools in the literature for the study of FIFO systems, attributing to the difficulty of the problem.
- **January 2022 - March 2022:** Apply our framework and use formal methods to verify web services orchestration and choreography.
- **January 2022 - June 2022:** Writing of the thesis.

5 Curriculum Vitae

5.1 List of contributions

- *Bounded Reachability Problems are Decidable in FIFO Machines.* B. Bollig, A. Finkel, and A. Suresh. In Proceedings of the 31st International Conference on Concurrency Theory (CONCUR'20), volume 171 of Leibniz International Proceedings in Informatics, Vienna, Austria, September 2020. *Received Best Paper Award.*

Submitted

- *A Unifying Framework for Deciding Synchronizability.* B.Bollig, C. Di Guisto, A. Finkel, L. Laversa, E. Lozes, A. Suresh. Submitted to CONCUR 2021.
- *Bounded Reachability Problems are Decidable in FIFO Machines.* B. Bollig, A. Finkel, and A. Suresh. Extended version submitted to LMCS.

In progress

- *Verification of Branch Well Structured Transition Systems.* B. Bollig, A. Finkel, and A. Suresh, 2021.
- *Reachability is Undecidable in Send-Synchronizable Systems.* B. Bollig, A. Finkel, and A. Suresh, 2021.

5.2 Training for ED-STIC

Scientific

- MOVEP Summer School 2020 - 36 hours
- Advanced Course on Automata, Logic and Games 2021 - In progress

Non-scientific

- Français langue étrangère: niveau B2- 25 hours
- Ethics and STIC - 12 hours
- Programme de mentorat pour les doctorantes - 25 hours
- Writing a scientific paper - 4 hours

References

[1] Parosh Aziz Abdulla, Aurore Collomb-Annichini, Ahmed Bouajjani, and Bengt Jonsson. Using forward reachability analysis for verification of lossy channel systems. *Formal Methods Syst. Des.*, 25(1):39–65, 2004.

[2] C. Aiswarya, Paul Gastin, and K. Narayan Kumar. Verifying communicating multi-pushdown systems via split-width. In *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014*, volume 8837 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2014.

[3] Mohamed Faouzi Atig, Benedikt Bollig, and Peter Habermehl. Emptiness of ordered multi-pushdown automata is 2etime-complete. *Int. J. Found. Comput. Sci.*, 28(8):945–976, 2017.

[4] Samik Basu and Tevfik Bultan. Choreography conformance via synchronizability. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 795–804. ACM, 2011.

[5] Samik Basu, Tevfik Bultan, and Meriem Ouederni. Deciding choreography realizability. In John Field and Michael Hicks, editors, *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, January 22-28, 2012*, pages 191–202. ACM, 2012.

[6] Marcello M. Bersani and Stéphane Demri. The complexity of reversal-bounded model-checking. In Cesare Tinelli and Viorica Sofronie-Stokkermans, editors, *Frontiers of Combining Systems, 8th International Symposium, FroCoS 2011, Saarbrücken, Germany, October 5-7, 2011. Proceedings*, volume 6989 of *Lecture Notes in Computer Science*, pages 71–86. Springer, 2011.

[7] Benedikt Bollig and Paul Gastin. Non-sequential theory of distributed systems. *CoRR*, abs/1904.06942, 2019.

[8] Ahmed Bouajjani, Constantin Enea, Kailiang Ji, and Shaz Qadeer. On the completeness of verifying message passing programs under bounded asynchrony. In Hana Chockler and Georg Weissenbacher, editors, *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Proceedings, Part II*, volume 10982 of *Lecture Notes in Computer Science*, pages 372–391. Springer, 2018.

[9] Daniel Brand and Pitro Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, 1983.

[10] Gérard Cécé and Alain Finkel. Verification of programs with half-duplex communication. *Inf. Comput.*, 202(2):166–190, 2005.

[11] Pierre Chambart and Philippe Schnoebelen. The ordinal recursive complexity of lossy channel systems. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008*, pages 205–216. IEEE Computer Society, 2008.

[12] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[13] Bruno Courcelle. Special tree-width and the verification of monadic second-order graph properties. In *FSTTCS*, volume 8 of *LIPIcs*, pages 13–29, 2010.

[14] Aiswarya Cyriac, Paul Gastin, and K. Narayan Kumar. MSO decidability of multipushdown systems via split-width. In Maciej Koutny and Irek Ulidowski, editors, *CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4-7, 2012. Proceedings*, volume 7454 of *Lecture Notes in Computer Science*, pages 547–561. Springer, 2012.

[15] Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for petri nets is not elementary. In *Proceedings*

of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, pages 24–33. ACM, 2019.

[16] 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*, pages 285–294. IEEE Computer Society, 2012.

[17] 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.

[18] Alain Finkel. Decidability of the termination problem for completely specified protocols. *Distributed Computing*, 7(3):129–135, 1994.

[19] 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.

[20] Alain Finkel and Étienne Lozes. Synchronizability of communicating finite state machines is not decidable. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 122:1–122:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[21] 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*, volume 140 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[22] Alain Finkel and Arnaud Sangnier. Reversal-bounded counter machines revisited. In Edward Ochmanski and Jerzy Tyszkiewicz, editors, *Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008, Torun, Poland, August 25-29, 2008, Proceedings*, volume 5162 of *Lecture Notes in Computer Science*, pages 323–334. Springer, 2008.

[23] Blaise Genest, Dietrich Kuske, and Anca Muscholl. On communicating automata with bounded channels. *Fundam. Inform.*, 80(1-3):147–167, 2007.

[24] Blaise Genest, Dietrich Kuske, and Anca Muscholl. On communicating automata with bounded channels. *Fundamenta Informaticae*, 80(1-3):147–167, 2007.

[25] Blaise Genest, Anca Muscholl, and Dietrich Kuske. A kleene theorem for a class of communicating automata with effective algorithms. In Cristian Calude, Elena Calude, and Michael J. Dinneen, editors, *Developments in Language Theory, 8th International Conference, DLT 2004, Auckland, New Zealand, December 13-17,*

2004, Proceedings, volume 3340 of *Lecture Notes in Computer Science*, pages 30–48. Springer, 2004.

[26] Cinzia Di Giusto, Laetitia Laversa, and Étienne Lozes. On the k-synchronizability of systems. In Jean Goubault-Larrecq and Barbara König, editors, *Foundations of Software Science and Computation Structures - 23rd International Conference, FOSSACS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings*, volume 12077 of *Lecture Notes in Computer Science*, pages 157–176. Springer, 2020.

[27] 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.

[28] Jesper G. Henriksen, Madhavan Mukund, K. Narayan Kumar, Milind Sohoni, and P.S. Thiagarajan. A theory of regular msc languages. *Information and Computation*, 202(1):1–38, 2005.

[29] Alexander Heußner, Jérôme Leroux, Anca Muscholl, and Grégoire Sutre. Reachability analysis of communicating pushdown systems. In C.-H. Luke Ong, editor, *Foundations of Software Science and Computational Structures, 13th International Conference, FOSSACS 2010*, volume 6014 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2010.

[30] Oscar H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *J. ACM*, 25(1):116–133, 1978.

[31] Thierry Jéron and Claude Jard. Testing for unboundedness of FIFO channels. *Theor. Comput. Sci.*, 113(1):93–117, 1993.

[32] Dietrich Kuske and Anca Muscholl. Communicating automata, 2014.

[33] Salvatore La Torre, P. Madhusudan, and Gennaro Parlato. Context-bounded analysis of concurrent queue systems. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008*, volume 4963 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 2008.

[34] Julien Lange and Nobuko Yoshida. Verifying asynchronous interactions via communicating session automata. *CoRR*, abs/1901.09606, 2019.

[35] Markus Lohrey and Anca Muscholl. Bounded MSC communication. In Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and Computation Structures, 5th International Conference, FOSSACS 2002. Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2002 Grenoble, France, April 8-12, 2002, Proceedings*, volume 2303 of *Lecture Notes in Computer Science*, pages 295–309. Springer, 2002.

[36] Markus Lohrey and Anca Muscholl. Bounded MSC communication. *Inf. Comput.*, 189(2):160–181, 2004.

[37] P. Madhusudan and Gennaro Parlato. The tree width of auxiliary storage. In *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2011*, pages 283–294. ACM, 2011.

[38] Ernst W. Mayr. An algorithm for the general petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984.

[39] Gérard Memmi and Alain Finkel. An introduction to fifo nets-monogeneous nets: A subclass of fifo nets. *Theor. Comput. Sci.*, 35:191–214, 1985.

[40] Bernard Vauquelin and Paul Franchi-Zannettacci. Automates a file. *Theor. Comput. Sci.*, 11:221–225, 1980.

[41] Gregor von Bochmann. Communication protocols and error recovery procedures. *Operating Systems Review*, 9(3):45–50, 1975.

[42] 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.