Challenges of the Reachability Problem in Infinite-State Systems

Wojciech Czerwiński
University of Warsaw, Poland

Abstract

The reachability problem is a central problem for various infinite state systems like automata with pushdown, with different kinds of counters or combinations thereof. Despite its centrality and decades of research the community still lacks a lot of answers for fundamental and basic questions of that type. I briefly describe my personal viewpoint on the current state of art and emphasise interesting directions, which are worth investigating in my opinion. I also formulate several easy to formulate and understand challenges, which might be pretty hard to solve but at the same time illustrate fundamental lack of our understanding in the area.

2012 ACM Subject Classification Theory of computation → Parallel computing models

Keywords and phrases reachability problem, infinite-state systems, vector addition systems, pushdown

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.2

Category Invited Paper

Funding *Wojciech Czerwiński*: Supported by the ERC grant INFSYS, agreement no. 950398.

Introduction

Since the early days of computer science the reachability problem is in its core interest. One of the first known undecidable problems was the halting problem for Turing Machines (TM), which is actually exactly the reachability problem for TM. We are given an initial configuration and final configuration and we ask whether there is a path of the TM between them. It is quite easy to come up with simpler automata models for which the reachability problem is also undecidable. The undecidability will follow from the fact that the simpler models can actually simulate TM, so are Turing powerful.

The first such a model is automaton with two pushdowns or in other words with two stacks. It has two stacks and each transition can change a state and push or pop on the top of the stacks. One can easily see that the transition of a TM can be simulated by automaton with two stacks. Let us say that the TM has one tape. One stack keeps the part of the tape of the TM to the left of the head and the other stack keeps the part of the tape to the right of the head. The top elements of both stacks are the cells of the TM, which are close to its head. To simulate a transition of a TM, say one step of the head to the left, the first stack pops the top symbol and the second stack pushes the same symbol. It is rather straightforward to see that this simulation indeed works well, which shows that automata with two stacks have undecidable reachability problem.

One can however simplify further and show that the reachability problem is undecidable already for the automaton with two counters, which can be zero-tested. Configuration of such an automaton consists of a finite state and two nonnegative integer counters. A transition can change the state and increase or decrease the counters. One can assume that increases and decreases are just by one or by some arbitrary value, it does not change the strength of the model. There are also special transitions called zero-tests. Such a transition can be fired only if a specific counter (the first or the second one) equals exactly zero. This model was shown to be undecidable by Marvin Minsky [19], so it is sometimes called a Minsky machine.

The reduction from the automaton with two stacks to the automaton with two zero-tested counters is not very complicated. A very brief sketch is the following: one stack can be simulated easily by two counters with zero-tests, one keeping value of the stack as a B-ary number (where B is the number of different stack symbols) and the other one being auxiliary. Indeed, with the help of an auxiliary counter with zero-tests one can multiply or divide the value of the first counter by B and this roughly speaking corresponds to a push or pop of a symbol from the top of the stack. Thus one can simulate automaton with two stacks by an automaton with four counters with zero-tests. Further, one can observe that the auxiliary counter can be shared, so only three counters are needed for the simulation of two stacks. Finally, one can simulate values of three counters x, y and z on two counters, one counter keeping value $2^x 3^y 5^z$ and the other one being an auxiliary counter, which helps simulating incrementing and decrementing of the three counters. The above construction can be easily made precise, and it shows that the reachability problem for automata with two zero-tested counters is indeed undecidable.

Decidable models

Automata models with undecidable reachability problem are of not much use: even the simplest fundamental problem cannot be decided for them. Therefore in order to understand the computation better, the infinite-state community searched for decidable restrictions of the above models. Of course there is plenty of ways one can restrict pushdowns or counters, but there are a few natural ones, which were considered for decades. Quite natural restrictions are in my opinion the following:

(1) disallow zero-tests and consider just nonnegative integer counters without zero-tests
(2) simplify the counters even further and allow them go to below zero
(3) consider nonnegative integer counters with just one counter being zero-tested or some other restricted version of zero-tests
(4) consider one stack with possible other counters (nonnegative integer lub just integer), but without zero-tests

Of course one can come up with other models, but it turns out that the above models were widely considered.

Below we briefly discuss all the four options and also some other interesting models. Along the way I share with you my personal feelings what seems to be interesting in my opinion. I also list several challenges, which are easy to formulate, but I think require new understanding of the models we work with. Solving them might be very hard or maybe easy, but definitely would lead to new insights in the area.

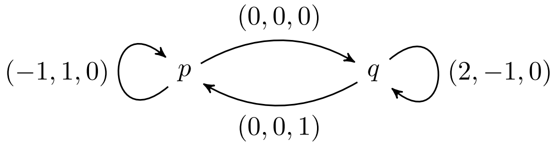
Vector addition systems

An automaton with d counters, which cannot be zero-tested on each transition just increments or decrements its counters by fixed values. It can be seen as an automaton adding vectors from $\mathbb{Z}^d$. Indeed, a configuration of such an automaton consists of a state and values of the d nonnegative integer counters, which can be seen as a vector in $\mathbb{N}^d$. A transition just adds a vector in $\mathbb{Z}^d$ to the current vector, it can be fired if the result of the addition is still a vector in $\mathbb{N}^d$. Such automata are called Vector Addition Systems with States, shortly VASS. If there is just one state they are called Vector Addition Systems, shortly VAS. It turns out that the reachability problems for VASS, for VAS and for Petri nets, a popular model of parallel computation are easily interreducible. The most robust model of out them is, in my opinion, the VASS model, so I will refer to the reachability problem in VASS. The reachability problem was shown decidable in 1981 [18], but only recently its complexity was settled to be Ackermann-complete. The upper bound was shown in 2019 [15], while the Ackermann-hardness in 2021 [6, 17]. The complexity class Ackermann is roughly speaking the class of problems, which can be solved in time around the Ackermann function. One can define easily the Ackermann function (some version of it, there are different versions, but they do not change the definition of the Ackermann complexity class as explained in [21]) using the so called hierarchy of fast growing functions F_k. We write $F_1(n) = 2n$ and $F_k(n) = \underbrace{F_{k-1} \circ \ldots \circ F_{k-1}}_{n}(1)$ for any $k \geq 2$. One can see that in particular $F_2(n) = 2^n$ and $F_3(n) = \text{Tower}(n)$. Then we define the Ackermann function $A(n) = F_n(n)$. Notice that $A(1) = 2$, $A(2) = 4$, $A(3) = 16$, while already $A(4) = \text{Tower}(\text{Tower}(\text{Tower}(\text{Tower}(1)))) = \text{Tower}(\text{Tower}(\text{Tower}(2))) = \text{Tower}(\text{Tower}(4)) = \text{Tower}(65536)$, which is an incredibly huge number. It is hard to imagine how fast the Ackermann function grows. Based on the F_k functions one can define the complexity classes $\mathcal{F}_k$, which contain roughly speaking problems solvable in time around F_k of the input size. Details can be found here [21].

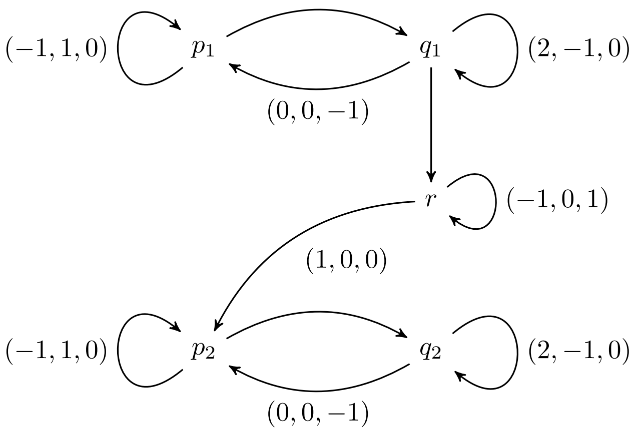
Despite the fact that the complexity of the reachability problem for VASS have been settled we still lack a lot of knowledge about the reachability properties in VASS. This can be easily seen if we look into VASS with d counters, which we also call d-dimensional VASS or shortly d-VASS. In general the complexity of the reachability problem for d-VASS is known to be in $\mathcal{F}_d$ [10]. The best currently known lower bound is $\mathcal{F}_d$-hardness in dimension $2d + 3$ [4]. The gap does not look big till we look at d-VASS for some small fixed d. Notice however, that the mentioned bounds give us Tower upper complexity bound for 3-VASS, but Tower-hardness only for 9-VASS, which is quite a big gap in understanding.

Let us summarise what is known about the reachability problem for d-VASS for small dimensions d. For small complexities it matters whether the numbers on transitions are represented in unary or in binary. Clearly binary representation is more concise, so possibly complexity of any problem for VASS with numbers represented in binary (shortly binary VASS) can be higher than the complexity for VASS with numbers represented in unary (shortly unary VASS). For dimension one it is rather easy to see that if there is a path in unary 1-VASS then there is also a path of at most cubic length, which easily implies NL algorithm and therefore also NL-completeness of the reachability problem. The complexity for binary 1-VASS is known to be NP-complete since 2009 [13]. A few years later came a big progress for 2-VASS. For binary 2-VASS it was shown in [2] that if there is a path between two configurations then there is also a path of at most exponential length. This easily implies existence of a PSpace algorithm, which just guesses this path step by step. Altogether the problem was shown in [2] to be PSpace-complete. One year later authors of [9] have proved that in unary 2-VASS if there is a path between two configurations then there is also a path of at most polynomial length, which implies algorithm in NL and thus NL-completeness.

It is a major challenge to go beyond dimension two. One particular reason for that is that reachability sets (and actually relations as well) are semilinear in 2-VASS, while it is not true for 3-VASS [14]. Recall that a set is linear of it is of a form $b + P^*$ for some vector $b \in \mathbb{N}^d$ and finite set $P \subseteq \mathbb{N}^d$ and a set is semilinear if it is a finite union of linear sets. The following example of a 3-VASS is probably the simplest way to see this and comes from the work of Hopcroft and Pansiot [14], thus we call it the HP-example. One can check that the set of configurations reachable from $p(1, 0, 0)$ in state p is of a form $\{p(x, y, z) \mid 1 \leq x + y \leq 2^z\}$, which is clearly not semilinear. Another challenge is that already in dimension three one can get VASS with finite reachability set, but of size k-fold exponential, so it is hard to obtain complexity bound below the size of the finite reachability set, so below Tower. To see that



such 3-VASS exist consider an example of a composition of a bit modified (transition from q to p now subtracts one, not adds one) two copies of the HP-example composed with itself. One can quite easily check that the reachability set from $p_1(1, 0, n)$ is doubly-exponential in n.



So see this notice that in particular any configuration of the form $p_2(x, y, 0)$ for $x + y = 2^{2^n}$ can be reached from $p_1(1, 0, n)$. Compositing k copies of the HP-example in the same fashion would result in getting $(k - 1)$-fold exponential size finite reachability set.

Unfortunately, not much upper or lower bounds are known for 3-VASS and for other d-VASS for low fixed $d \in \mathbb{N}$. I believe it is an important direction and I keep working in this area since a few years. We have some partial results with co-authors. In [5] we have shown that there exist a unary 3-VASS with two configurations such that the shortest path between them is of an exponential length. This distinguishes the situation from the unary 2-VASS, where the shortest path is always polynomial. Therefore the same way of proving membership in NL cannot work for unary 3-VASS. In fact recently together with Dmitry Chistikov, Filip Mazowiecki, Łukasz Orlikowski, Henry Sinclair-Banks and Karol Węgrzycki we have shown that the reachability problem for unary 3-VASS is NP-hard, it is a part of a paper just accepted to FOCS 2024.

On the other hand together with Ismaël Jecker, Sławomir Lasota and Łukasz Orlikowski we seem to have shown that in binary 3-VASS if there is a path between two configurations then there is one of at most doubly-exponential length. This would imply an ExpSpace membership for the reachability problem. However, the result is not yet written down, so there might be some mistake there. But, even if this result is true it might not be optimal. In every example of a 3-VASS we know, the shortest paths between two configurations are at most exponential, which would suggest PSpace algorithm. The lowest dimension in which we know that sometimes the shortest paths are of doubly-exponential length is four, in [5] we have shown an example of a 4-VASS with shortest path between some two configurations being doubly-exponential. Thus, I think that the following challenge is a very interesting one and could lead to getting a tight complexity bound for 3-VASS, which are still quite a simple and natural computation model.

▶ **Challenge 1.** *Find an example of a binary* 3*-VASS with doubly-exponential shortest path between two configurations.*

Another interesting challenge is to decide for which dimensions d reachability in d-VASS is elementary, so below Tower. The paper [4] gives us $\mathcal{F}_d$-hardness in dimension $2d + 3$, so Tower-hardness for 9-VASS. However, we have managed to show Tower-hardness already for 8-VASS in [7]. The question remains for d-VASS when $d \in \{4, 5, 6, 7\}$ (and also for $d = 3$ if our supposed ExpSpace membership for 3-VASS would turn wrong). Of course to get Tower-hardness for d-VASS we need to know examples of d-VASS with at least Tower-long shortest paths between two configurations. However, we even know no example below dimension eight with such a property. This leads to another interesting challenge.

▶ **Challenge 2.** *Find an example of a binary d-VASS with at least tower-long shortest path between two configurations for $d \leq 7$.*

Of course it might be that solving Challenges 1 and 2 is impossible, because there exist no such examples. This would however be probably hard to show, as designing faster algorithms is usually rather sophisticated. I list here challenges, which possibly might be resolved positively during one afternoon. That is why I have formulated the challenges in such a form.

Another interesting open problem for reachability in VASS is the question for fixed VASS. There is a conjecture that for every VASS V there is a constant C_V such that if there is a path from a configuration s to a configuration t then there is also one of length at most C_V times the sum of sizes of s and t [8]. Such a conjecture, if true, would imply a PSpace algorithm for the reachability problem for any fixed binary VASS V. The PSpace-hardness was already shown in [8]. I personally believe in this conjecture, but I might be wrong, so there is another hard challenge.

▶ **Challenge 3.** *Find an example of a binary VASS for which shortest paths in between its configurations are longer than linear.*

Integer VASS

Another way of relaxing infinite-state systems is to consider automata with integer counters, which cannot be zero-tested and moreover can drop below zero. Such systems are called $\mathbb{Z}$-VASS or integer VASS. Even though it is not entirely obvious, it is easy to reduce the reachability for $\mathbb{Z}$-VASS to the reachability for VASS. Every $\mathbb{Z}$-VASS counter can be simulated as a difference of two VASS counters. However, the reachability of $\mathbb{Z}$-VASS is much easier than the reachability for VASS, it is known to be NP-complete [12].

The reachability problem for automata with only $\mathbb{Z}$-counters have been well understood, as seen above. However, the power of adding $\mathbb{Z}$-counters to other systems is definitely not so fully understood. Actually, it is not clear whether an automaton with d VASS-counters and k integer-counters is closer in behaviour to a d-VASS or to $(d + k)$-VASS. One particularly interesting model is a 2-VASS with many $\mathbb{Z}$-counters. There are no known examples of these systems for which the shortest path is longer than exponential, so it is very possible that the reachability problem is in PSpace, thus PSpace-complete. On the other hand, to my best knowledge, there is no known upper complexity bound below Ackermann. Thus it is a natural challenge to investigate these systems.

▶ **Challenge 4.** *Find an example of a binary* 2*-VASS with additional $\mathbb{Z}$-counters with shortest path being longer than exponential.*

My personal conjecture is that there exists no system as mentioned above in Challenge 4, but I might be very wrong, it is always hard to imagine sophisticated examples. Notice that in the HP-example the third counter can be treated as a $\mathbb{Z}$-counter (as it never drops below zero), so reachability sets in 2-VASS with $\mathbb{Z}$-counters are not necessarily semilinear.

VASS with zero-tests

As mentioned at the beginning of this text, automata with two zero-tested counters already have undecidable reachability problem. But maybe we can allow for one zero-tested counter? If we have just an automaton with one zero-tested unary-encoded counter it is easy to show that if there is a path between two configurations then there is also a path of at most polynomial length. This means that the reachability problem is NL-complete for such automata. But, if we add more VASS-counters on top of the zero-tested counter then the situation is more sophisticated. However, in 2008 Klaus Reinhard have shown that the reachability problem for VASS with one zero-tested counter is decidable [20]. Actually, he has shown even more: decidability for VASS with hierarchical zero-tests. A d-VASS with hierarchical zero-tests (hierarchical VASS) have the following zero-tests: for each $k \leq d$ one can test whether all the first k counters are equal to zero at the same time. Interestingly, this model also has decidable reachability problem, even though it is so closed to the automaton with two zero-tested counters. Unfortunately, the proof of Reinhard was very complicated and hard to understand, so there were other tries to explain the nature of VASS with zero-tests. In 2011 Bonnet reproved the result using other techniques [3], but his work considered only VASS with one zero-test, not a hierarchical VASS. Very recently Guttenberg came up with another proof of decidability of the reachability problem for hierarchical VASS [11], this time hopefully more understandable.

Even though the decidability status of the reachability problem for hierarchical VASS is resolved its complexity is unclear. It cannot be better than Ackermann, as already the reachability problem for VASS is Ackermann-hard. However, it is interesting whether the problem is in Ackermann or maybe much higher. In particular, it would be interesting to know whether hierarchical zero-tests add anything to the power of VASS, namely whether hierarchical d-VASS are harder at some dimensions than the classical d-VASS. Interestingly, it was shown that in dimension two both models are very similar and the reachability problem for hierarchical 2-VASS is in PSpace [16]. Could this be true also in higher dimensions? It is an interesting challenge to find this out.

Pushdown VASS

Similarly as for zero-tests one can consider automata with just one stack. Reachability in a pushdown automaton is easy and can be performed in polynomial time, for example by the famous CYK algorithm. However, things become much harder if we add additional VASS-counters to this stack. An automaton with a stack and d VASS-counters is called a d-dimensional Pushdown VASS (d-PVASS). Similarly as pushdown automata are equivalent to context-free grammars the PVASS are equivalent to Grammar VAS (GVAS), which are just context-free grammars with terminals being vectors in $\mathbb{Z}^d$. A derivation of a GVAS is correct if starting in the initial vector in $\mathbb{N}^d$ and reading leaves in the prefix order one never drops below zero and finally reaches the target vector in $\mathbb{N}^d$.

Little is known about the reachability problem for PVASS. It is conjectured to be decidable, but there is no known algorithm even for 1-dimensional PVASS. An interesting result was shown by Atig and Ganty [1]. They have proved that for so called finite-index

GVAS the reachability problem is decidable. Concretely speaking they have reduced the reachability problem for finite-index d-GVAS with k nonterminals to the reachability problem for hierarchical dk-VASS, which is known to be decidable [20, 11].

Recently with Clotilde Bizière we have considered the reachability problem for 1-GVAS and we have an impression that we can show decidability by a reduction to the finite-index case. However, this result is also not confirmed and may contain a flaw. Even if it is correct the complexity of the problem for 1-GVAS remains open. We do not know any example of a 1-GVAS in which a path of length longer than exponential is needed, which would suggest low complexity. On the other hand even very simple 1-GVAS may have finite, but huge reachability sets. For example this 1-GVAS with just three nonterminals have a reachability set of Tower-size

- $X \longrightarrow 0$, $X \longrightarrow -1X2$; and
- $Y \longrightarrow 1$, $Y \longrightarrow -1YX$; and
- $Z \longrightarrow 1$; $Z \longrightarrow -1ZY$.

Indeed, with the starting nonterminal Z and starting value n we might reach up to Tower(n) values. To see this observe that nonterminal X with starting value n might output value up to $2n$. It turn the nonterminal Y with starting value n may produce up to n variables X and output value up to 2^n. Therefore nonterminal Z with starting value n may produce up to n variables Y and output value up to Tower(n). Similar grammars with d nonterminals may have finite reachability sets of size up to $F_d(n)$. So it looks interesting that we do not have any example of 1-GVAS with longer than exponential shortest path.

▶ **Challenge 5.** *Find an example of a* 1*-GVAS such that shortest derivations are longer than exponential.*

Other models

There are also other very interesting and natural models of infinite-state systems. One popular such system is Branching VAS (BVAS), for which the reachability problem is open in general, but was recently shown decidable in 2-dimensional BVAS. The results are not yet published, but they were referred here by Clotilde Bizière: `https://youtu.be/aKIxlgQAa2w?si=V7yWXsj8_JaoiBHv`.

Another interesting model, which illustrated how much we are in trouble with automata, which have both additive and multiplicative (or recursive) behaviour is a model about which I have heard from Anthony Lin. Consider a model, let us call it the Lin automaton, which is an automaton with one counter. A transition can increase or decrease this counter, as normally, but it can also multiply it or divide by two. Then it is unclear whether the reachability problem for such a simple automaton is decidable. Together with Moses Ganardi, Łukasz Orlikowski and Georg Zetzsche we think about this problem and we have some partial results. However, we still do not have any example of a Lin automaton, which has a path longer than exponential. Thus the following might be an interesting challenge.

▶ **Challenge 6.** *Find an example of a Lin-automaton such that shortest derivations are longer than exponential.*

To summarise, it seems that there are still many fundamental and natural challenges in the field of infinite-state systems, some of which are asking about deep properties of the computation. It might be that we are still at the beginning of our way to understand the computation even from this very high level perspective.

References

1 Mohamed Faouzi Atig and Pierre Ganty. Approximating Petri Net Reachability Along Context-free Traces. In *Proceedings of FSTTCS 2011*, volume 13 of *LIPIcs*, pages 152–163, 2011.

2 Michael Blondin, Alain Finkel, Stefan Göller, Christoph Haase, and Pierre McKenzie. Reachability in Two-Dimensional Vector Addition Systems with States Is PSPACE-Complete. In *Proceedings of LICS 2015*, pages 32–43. IEEE Computer Society, 2015.

3 Rémi Bonnet. The Reachability Problem for Vector Addition System with One Zero-Test. In *Proceedings of MFCS 2011*, volume 6907 of *Lecture Notes in Computer Science*, pages 145–157, 2011.

4 Wojciech Czerwinski, Ismaël Jecker, Slawomir Lasota, Jérôme Leroux, and Lukasz Orlikowski. New Lower Bounds for Reachability in Vector Addition Systems. In *Proceedings of FSTTCS 2023*, volume 284 of *LIPIcs*, pages 35:1–35:22, 2023.

5 Wojciech Czerwiński, Sławomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. Reachability in Fixed Dimension Vector Addition Systems with States. In *Proceedings of CONCUR 2020*, volume 171 of *LIPIcs*, pages 48:1–48:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

6 Wojciech Czerwiński and Łukasz Orlikowski. Reachability in Vector Addition Systems is Ackermann-complete. In *Proceedings of FOCS 2021*, pages 1229–1240. IEEE, 2021.

7 Wojciech Czerwiński and Łukasz Orlikowski. Lower Bounds for the Reachability Problem in Fixed Dimensional VASSes. In *Proceedings of LICS 2022*, pages 40:1–40:12. ACM, 2022.

8 Andrei Draghici, Christoph Haase, and Andrew Ryzhikov. Reachability in Fixed VASS: Expressiveness and Lower Bounds. In *Proceedings of FoSSaCS 2024*, volume 14575 of *Lecture Notes in Computer Science*, pages 185–205, 2024.

9 Matthias Englert, Ranko Lazic, and Patrick Totzke. Reachability in Two-Dimensional Unary Vector Addition Systems with States is NL-Complete. In *Proceedings of LICS 2016*, pages 477–484. ACM, 2016.

10 Yuxi Fu, Qizhe Yang, and Yangluo Zheng. Improved Algorithm for Reachability in d-VASS. In *Proceedings of ICALP 2024*, volume 297 of *LIPIcs*, pages 136:1–136:18, 2024.

11 Roland Guttenberg. Flattability of Priority Vector Addition Systems. In *Proceedings of ICALP 2024*, volume 297 of *LIPIcs*, pages 141:1–141:20, 2024.

12 Christoph Haase and Simon Halfon. Integer Vector Addition Systems with States. In *Reachability Problems – 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings*, volume 8762 of *Lecture Notes in Computer Science*, pages 112–124, 2014.

13 Christoph Haase, Stephan Kreutzer, Joël Ouaknine, and James Worrell. Reachability in Succinct and Parametric One-Counter Automata. In *Proceedings of CONCUR 2009*, pages 369–383. Springer Berlin Heidelberg, 2009.

14 John E. Hopcroft and Jean-Jacques Pansiot. On the Reachability Problem for 5-Dimensional Vector Addition Systems. *Theor. Comput. Sci.*, 8:135–159, 1979.

15 Jérôme Leroux and Sylvain Schmitz. Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension. In *Proceedings of LICS 2019*, pages 1–13. IEEE, 2019.

16 Jérôme Leroux and Grégoire Sutre. Reachability in Two-Dimensional Vector Addition Systems with States: One Test Is for Free. In *Proceedings of CONCUR 2020*, volume 171 of *LIPIcs*, pages 37:1–37:17, 2020.

17 Jérôme Leroux. The Reachability Problem for Petri Nets is Not Primitive Recursive. In *Proceedings of FOCS 2021*, pages 1241–1252, 2022.

18 Ernst W. Mayr. An Algorithm for the General Petri Net Reachability Problem. In *Proceedings of STOC 1981*, pages 238–246, 1981.

19 Marvin L. Minsky. *Computation: finite and infinite machines.* Prentice-Hall, Inc., USA, 1967.

20 Klaus Reinhardt. Reachability in Petri Nets with Inhibitor Arcs. *Electron. Notes Theor. Comput. Sci.*, 223:239–264, 2008.

21 Sylvain Schmitz. Complexity Hierarchies beyond Elementary. *ACM Trans. Comput. Theory*, 8(1):3:1–3:36, 2016.