Reachability in Register Machines with Polynomial Updates

Alain Finkel[1,⋆], Stefan Göller[2], and Christoph Haase[1,⋆]

[1] LSV - CNRS & ENS Cachan, France
{finkel,haase}@lsv.ens-cachan.fr
[2] Institut für Informatik, Universität Bremen, Germany
goeller@informatik.uni-bremen.de

Abstract. This paper introduces a class of register machines whose registers can be updated by polynomial functions when a transition is taken, and the domain of the registers can be constrained by linear constraints. This model strictly generalises a variety of known formalisms such as various classes of Vector Addition Systems with States. Our main result is that reachability in our class is PSPACE-complete when restricted to one register. We moreover give a classification of the complexity of reachability according to the type of polynomials allowed and the geometry induced by the range-constraining formula.

1 Introduction

Register machines are a class of abstract machines comprising a finite-state controller with a finite number of integer-valued registers that can be manipulated or tested when a transition is taken. A prominent instance are *counter machines* due to Minsky [18], which are obtained by restricting registers to range over the naturals, allowing for addition of integers to the registers along transitions, and testing registers for zero. A seminal result by Minsky states that counter machines are Turing powerful in the presence of at least two registers. Decidability can be obtained by further restricting counter machines and disallowing zero tests, which yields a class of register machines known as *Vector Addition Systems with States (VASS)* or *Petri nets*. Their reachability problem is known to be decidable and EXPSPACE-hard [17,16].

A number of extensions, generalisations and restrictions of VASS can be found in the literature. For instance, various extensions that increase the power of transitions have been studied, including Reset/Transfer (Petri) nets [6], Petri nets with inhibitory arcs [3], or Affine nets [8] which extend VASS such that transitions can be any non-decreasing affine function; any of these extensions lead to undecidability of reachability in the presence of more than one register. On the other hand, relaxing the domain of the registers of a VASS to the integers, or restricting VASS to just one register renders reachability NP-complete [12].

⋆ The authors are supported by the French Agence Nationale de la Recherche, ReacHard (grant ANR-11-BS02-001).

In summary, we can identify three parameters in which the aforementioned classes of register machines differ and which impact their expressiveness and the complexity of reachability: (1) the number of registers available, (2) the shape of the domain of the registers, and (3) the class of the transition functions used. In this paper, we generalise (3) and study the decidability and complexity of reachability when allowing for *polynomial functions with integer coefficients* to update register values. To this end, we introduce *polynomial register machines (PRMs)*, a class of register machines in which the previously mentioned classes of register machines embed smoothly. Of course, their undecidability results carry over, but on the positive side we are able to identify a decidable class of PRMs that is not contained in any of them.

The main result of this paper is to show that reachability in PRMs is PSPACE-complete when restricted to one register. As a motivating example, consider the question whether the following loop involving a single register variable x terminates:

```
int x := 0
while (x < 5):
  x := x**3 - 2x**2 - x + 2
```

This example is inspired by an example given in [1], and in this example x alternates between 0 and 2, and thus the loop never terminates. In fact, it is not difficult to see that the loop never terminates for all values $x < 3$. However for polynomials of higher degree and loops with a richer control structure, deciding termination becomes non-obvious. Even in dimension one, problems of this nature can become intriguingly difficult, see *e.g.* [2] for a discussion on open problems of this kind. Reachability for non-deterministically applied affine transformations from a finite set in dimension one has been shown to be decidable in 2-EXPTIME by Fremont [9].

There are a number of obstacles making it challenging to show decidability and complexity results for reachability in PRMs. In some classes of register machines, semi-linearity of the reachability set can be exploited in order to show decidability. However, taking a single-state PRM with one self-loop that updates the only register x with the polynomial $p(x) = x^2$, we see that the reachability set is not semi-linear. Moreover, the representation of the values that the register x can take grows exponentially with the number of times the self-loop is taken, which makes it not obvious how to decide reachability in polynomial space only.

The property that the reachability set is not semi-linear separates languages generated by PRMs from classes of machines that have semi-linear reachability sets, such as VASS in dimension one. More interestingly, PRMs can generate languages that cannot be generated by general VASS, which do also not have semi-linear reachability sets: the language $L = \{a^{n^2} : n > 0\}$ over the singleton alphabet $\{a\}$ can easily be generated by a PRM with two control locations, but not by any VASS [15].

Besides the aforementioned related work, as indicated by the example above, work related to ours can be found in the area of program verification. In [1], Babić *et al.* describe a semi decision procedure for proving termination of loops

involving polynomial updates, similar to the one above. Another example is the work by Bradley *et al.* [4] which provides a semi decision procedure for so-called multipath polynomial programs. However, to the best of our knowledge, no sound and complete algorithm for problems of this kind exists.

Due to space constraints, we had to omit some proof details. An extended version of this paper containing the omitted proofs in an appendix can be obtained from the authors.

2 Preliminaries

Before we formally introduce PRMs, we provide some technical definitions and known results on elementary algebra and number theory.

2.1 Technical Definitions and Known Results

By $\mathbb{N}, \mathbb{Z}$, $\mathbb{R}$ and $\mathbb{C}$ we denote the naturals, integers, reals and complex numbers, respectively. All integers in this paper are assumed to be encoded in binary unless stated otherwise. For $z \in \mathbb{Z}$, we denote by $\operatorname{sgn} z$ the *sign of* z, and by $|z|$ its absolute value. For $r_1 \leq r_2 \in \mathbb{R}$, we denote by $[r_1, r_2]$ the *closed interval* $\{r \in \mathbb{R} : r_1 \leq r \leq r_2\}$.

By $\mathbb{Z}[\overline{x}]$ we denote the ring of polynomials with integer coefficients over variables $\overline{x} = (x_1, \ldots, x_n)$. A polynomial $p(x) \in \mathbb{Z}[x]$ will be written as $p(x) = a_n x^n + \cdots + a_1 x + a_0$, and represented in sparse encoding by a sequence of pairs $(i, a_i)_{i \in I}$, where $I \subseteq \{0, \ldots, n\}$ contains those indexes for which $a_i \neq 0$. Given $z \in \mathbb{Z}$ and $p(x)$ in our representation, deciding $p(z) > 0$ is known to be computable in polynomial time [5]. Given a root $c \in \mathbb{C}$ of $p(x)$, we will make use of the following bound from [14] on the magnitude of c:

$$|c| \leq 1 + \sum_{0 \leq i < n} |a_i / a_n|. \tag{1}$$

Recall that for all $m > 0$, $p(a) \equiv p(b) \bmod m$ whenever $a \equiv b \bmod m$ for all $m > 0$, *i.e.* all $p(x) \in \mathbb{Z}[x]$ are invariant w.r.t. residual classes. Given pairwise co-prime $m_1, \ldots, m_k > 0$ and $b_1, \ldots, b_k \in \mathbb{Z}$, the *Chinese remainder theorem* states that a system of k linear congruences $x \equiv b_i \bmod m_i$, $1 \leq i \leq k$ has a unique solution modulo $m_1 m_2 \cdots m_k$. Moreover, recall that the prime number theorem states that the number $\pi(n)$ of primes below n grows as $\pi(n) \sim n / \ln n$. In particular, this implies that $O(\log n)$ bits are sufficient to represent the n-th prime number.

A *linear constraint* $\phi(\overline{x})$ is a conjunction of atoms of the form $\top$ and $p(\overline{x}) \sim z$, where $p \in \mathbb{Z}[\overline{x}]$ is linear, $z \in \mathbb{Z}$ and $\sim \in \{<, \leq, =, \geq, >\}$. The set of *solutions of* $\phi(\overline{x})$ is $\{\overline{z} \in \mathbb{Z}^d : \phi[\overline{z}/\overline{x}] \text{ is true}\}$ that we also denote by $[\![\phi(\overline{x})]\!]$. We say that $[\![\phi(\overline{x})]\!]$ is *upward closed* if whenever $\overline{z} \in [\![\phi(\overline{x})]\!]$ then $\overline{z'} \in [\![\phi(\overline{x})]\!]$ for all $\overline{z'}$ such that $\overline{z} \preceq \overline{z'}$. Here, $\preceq$ denotes the natural component-wise extension of the order $\leq$ on $\mathbb{Z}$ to tuples over $\mathbb{Z}$.

2.2 Polynomial Register Machines

This section introduces polynomial register machines. We only give full definitions for dimension one, since the major part of this paper in Section 3 focuses on this class. From the definitions below, it is easy to generalise to higher dimensions, which we are only going to discuss briefly in Section 4.

A *polynomial register machine (PRM)* is a tuple $\mathcal{R} = (Q, \Delta, \lambda, \phi)$, where Q is a finite set of *states* or *control locations*, $\Delta \subseteq Q \times Q$ is the *transition relation*, $\lambda : \Delta \to \mathbb{Z}[x]$ is the *transition labelling function*, labelling each transition with an *update polynomial*, and $\phi(x)$ is a *global invariant,* which is a linear constraint. As a convention, we assume $0 \in [\![\phi(x)]\!]$, though all results in this paper hold without this assumption. We write $q \xrightarrow{p(x)} q'$ whenever $(q, q') \in \Delta$ and $\lambda(q, q') = p(x)$. The set $C(\mathcal{R})$ of *configurations of* $\mathcal{R}$ is $C(\mathcal{R}) \stackrel{\text{def}}{=} Q \times [\![\phi(x)]\!] \subseteq Q \times \mathbb{Z}$, and we write configurations in $C(\mathcal{R})$ as $q(z)$. The size $|\mathcal{R}|$ of $\mathcal{R}$ is the number of bits required to write down $\mathcal{R}$ and $P(\mathcal{R})$ denotes *the set of polynomials that occur in* $\mathcal{R}$.

The semantics of $\mathcal{R}$ is given by a transition system $T(\mathcal{R}) = (C(\mathcal{R}), \to_{\mathcal{R}})$, where $q(z) \to_{\mathcal{R}} q'(z')$ if $q \xrightarrow{p(x)} q'$ and $z' = p(z)$. *Reachability* is to decide, given $q, q' \in Q$ and $z, z' \in \mathbb{Z}$, does $q(z) \to^*_{\mathcal{R}} q'(z')$ hold? Clearly, this problem can be reduced in logarithmic space to deciding $q(0) \to^*_{\mathcal{R}'} q'(0)$ for some PRM $\mathcal{R}'$ linear in the size of $\mathcal{R}$, z and z'.

For dimensions $d > 1$, a d-PRM is obtained by amending the above definitions such that $\phi(\overline{x})$ is free in $\overline{x} = (x_1, \ldots, x_d)$ and transitions are labelled with vectors of polynomials $(p_1(x_1), \ldots, p_d(x_d))$ that are applied componentwise. The next example shows how some of the classes of register machines mentioned in the introduction can be embedded into PRMs.

Example 1. A dimension d-VASS is a d-PRM with global invariant $\phi(\overline{x}) = \bigwedge_{1 \leq i \leq d} x_i \geq 0$ and transition polynomials of the form $p_i(x_i) = x_i + a_i$; a bounded d-counter automaton [13] with bounds $\overline{b} = (b_1, \ldots, b_d) \in \mathbb{N}^d$ is a d-PRM with the same transition polynomials and $\phi(\overline{x}) = \bigwedge_{1 \leq i \leq d}(x_i \geq 0 \wedge x_i \leq b_i)$. A reset d-VASS [6] can be simulated by employing polynomials of the form $p_i(x_i) = 0$ for resets.

The previous examples lead us to a classification of update polynomials. We call a polynomial of the form $p(x) = a_1 x + a_0$ a *counter polynomial* if $a_1 = 1$, *counter-like polynomial* if $a_1 \in \{-1, 1\}$, and if the degree of $p(x)$ is one then $p(x)$ is called an *affine polynomial.*

3 Reachability for One Register

This section proves the main theorem of this paper and shows that reachability in PRMs is decidable and PSPACE-complete. For the lower bound, we show that reachability becomes PSPACE-hard for update polynomials of degree two, even if the global invariant is unconstrained and thus upward closed. Subsequently, we show a matching upper bound which involves a thorough analysis of paths in the transition systems generated by PRMs.

3.1 Hardness for PSPACE

We reduce from the reachability problem for *linear-bounded automata (LBA)*, which is a well-known PSPACE-complete problem. An LBA (without input alphabet) is a tuple $\mathcal{M} = (Q_{\mathcal{M}}, \Gamma, \Delta_{\mathcal{M}})$, where $Q_{\mathcal{M}}$ is a finite set of *states* and Γ is a finite *tape alphabet* implicitly containing two distinguished symbols $\triangleright$ and $\triangleleft$ acting as left delimiter ($\triangleright$) and right delimiter ($\triangleleft$). The *transition relation* is a relation $\Delta_{\mathcal{M}} \subseteq Q_{\mathcal{M}} \times \Gamma \times Q_{\mathcal{M}} \times \Gamma \times \{\leftarrow, \rightarrow\}$ such that $(q, \gamma, q', \gamma', d) \in \Delta_{\mathcal{M}}$ implies that whenever $\mathcal{M}$ is in state q reading γ at the current head position on the tape then $\mathcal{M}$ switches to the state q' writing γ' onto the tape and moving the head in direction $d \in \{\leftarrow, \rightarrow\}$. We assume $\Delta_{\mathcal{M}}$ to be constrained such that it respects the delimiters, *i.e.*, it fulfils the conditions

(i) $(q, \triangleright, q', \gamma, d) \in \Delta_{\mathcal{M}}$ implies $\gamma = \triangleright$ and $d = \rightarrow$; and
(ii) $(q, \triangleleft, q', \gamma, d) \in \Delta_{\mathcal{M}}$ implies $\gamma = \triangleleft$ and $d = \leftarrow$.

A *configuration of* $\mathcal{M}$ is a tuple $(q, \triangleright w \triangleleft, i)$, where $q \in Q$ is the current state, $w \in (\Gamma \setminus \{\triangleright, \triangleleft\})^*$ is the *tape content* and $i \in \{0, |w| + 1\}$ is the position of the read-write head. Hence, at head position 0 the tape content is $\triangleright$ and at position $|w| + 1$ it is $\triangleleft$. The *successor relation* $\rightarrow_{\mathcal{M}}$ between two configurations is defined in the standard way.

Deciding whether $(q_0, \triangleright 0^n \triangleleft, 0) \rightarrow^*_{\mathcal{M}} (q_f, \triangleright 0^n \triangleleft, 0)$ for given $n \in \mathbb{N}$ (in unary) and given states $q_0, q_f \in Q_{\mathcal{M}}$ of a given LBA $\mathcal{M}$ working on the alphabet $\{\triangleright, 0, 1, \triangleleft\}$ is well-known to be PSPACE-complete. For our reduction, let us fix such an LBA $\mathcal{M}$ and $n \in \mathbb{N}$. The goal of the remainder of this section is to show how we can compute in polynomial time from $\mathcal{M}$ and n a PRM $\mathcal{R} = (Q, \Delta, \lambda, \top)$ with particular control locations $q_{\mathcal{R}}, q'_{\mathcal{R}}$ such that $(q_0, \triangleright 0^n \triangleleft, 0) \rightarrow^*_{\mathcal{M}} (q_f, \triangleright 0^n \triangleleft, 0)$ if, and only if, $q_{\mathcal{R}}(0) \rightarrow^*_{\mathcal{R}} q'_{\mathcal{R}}(0)$, which gives PSPACE-hardness of reachability in PRMs.

To begin with, let us discuss an encoding of configurations of $\mathcal{M}$. In the following, let p_i denote the $(i + 3)$-th prime number, *i.e.*, $p_1 = 7$, $p_2 = 11$, $p_3 = 13$, *etc.* Recall that by the prime number theorem p_i can be represented using $O(\log i)$ bits. Set $P \stackrel{\text{def}}{=} \prod_{1 \leq i \leq n} p_i$, we call a residue class r modulo P *valid* if for each $1 \leq i \leq n$ there is some $b_i \in \{0, 1\}$ such that $r \equiv b_i \bmod p_i$. Otherwise, r is called *invalid*. Our idea is to encode a tape configuration $\triangleright w \triangleleft$ of $\mathcal{M}$ with $w = w_1 \cdots w_n \in \{0, 1\}^n$ via the unique valid residue class r modulo P satisfying $r \equiv w_i \bmod p_i$ for all $1 \leq i \leq n$. Consequently, we can establish a one-to-one correspondence between valid residue classes modulo P and tape contents of $\mathcal{M}$. Thus, modulo each prime p_i, we naturally view the residue classes 0 and 1 to encode the Boolean values 0 and 1, respectively. During the simulation of $\mathcal{M}$ by $\mathcal{R}$, we will need a way to remember that an error has occurred. For that reason, we extend the set of valid residue classes to the set S of *sane residue classes modulo* P. Let $0 \leq r < P$, we call r *sane* if for every $1 \leq i \leq n$ there is some $b_i \in \{0, 1, 2\}$ such that $r \equiv b_i \bmod p_i$. We regard the residue class 2 as *erroneous*. Finally, let us introduce some additional notation that allows us to alter a residue class r *locally*. Let $0 \leq r < P$, $1 \leq i \leq n$ and $0 \leq a < p_i$, we

$$\mathit{flip}_i(r) \stackrel{\text{def}}{=} \begin{cases} r[1 \bmod p_i] & \text{if } r \equiv 0 \bmod p_i \\ r[0 \bmod p_i] & \text{if } r \equiv 1 \bmod p_i \\ r[2 \bmod p_i] & \text{if } r \equiv 2 \bmod p_i \end{cases} \qquad \mathit{eqzero}_i(r) \stackrel{\text{def}}{=} \begin{cases} r[0 \bmod p_i] & \text{if } r \equiv 0 \bmod p_i \\ r[2 \bmod p_i] & \text{if } r \equiv 1 \bmod p_i \\ r[2 \bmod p_i] & \text{if } r \equiv 2 \bmod p_i \end{cases}$$

Fig. 1. The mappings flip_i and eqzero_i.

denote by $r[a \bmod p_i]$ the unique residue class r' modulo P satisfying

$$r' \equiv a \bmod p_i; \text{ and}$$
$$r' \equiv r \bmod p_j \text{ for all } 1 \leq j \leq n \text{ such that } j \neq i.$$

The existence of r' is guaranteed by the Chinese remainder theorem.

For each $1 \leq i \leq n$, we define mappings $\mathit{flip}_i, \mathit{eqzero}_i, \mathit{eqone}_i : S \to S$ that allow us to perform tests and operations on sane residue classes. The definitions of flip_i and eqzero_i are given in Figure 1. Given $0 \leq r < P$, $\mathit{flip}_i(r)$ flips the bit encoded in the residue class modulo p_i, provided it is not erroneous. If it is erroneous, it remains so after an application of flip_i. Similarly, eqzero_i allows for "guess-testing" of the bit encoded in the residue class modulo p_i: if $r \equiv 0 \bmod p_i$ then this value is preserved by the application of eqzero_i. Otherwise, eqzero_i maps r to 2 so that it informally speaking "remembers" the wrong guess by mapping to a value r' such that $r' \equiv 2 \bmod p_i$. The mapping eqone_i is defined analogously to eqzero_i and allows for "guess-testing" whether $r \equiv 1 \bmod p_i$. The crucial point of our reduction is that flip_i, eqzero_i and eqone_i can be implemented via quadratic polynomials with coefficients of polynomial bit size.

Lemma 2. *For any $1 \leq i \leq n$ and any of $\mathit{flip}_i, \mathit{eqzero}_i, \mathit{eqone}_i : S \to S$, there is a quadratic polynomial with coefficients from $\{0, \ldots, P-1\}$ that realises the respective function.*

Proof. Let us first give polynomials for each of the mappings that work in $\mathbb{Z}/p_i\mathbb{Z}$. One easily verifies that the polynomials

$$p_{\mathit{eqzero}}(x) \stackrel{\text{def}}{=} -x^2 + 3x \qquad p_{\mathit{flip}}(x) \stackrel{\text{def}}{=} 3 \cdot 2^{-1} \cdot x^2 - 5 \cdot 2^{-1} \cdot x + 1$$
$$p_{\mathit{eqone}}(x) \stackrel{\text{def}}{=} x^2 - 2x + 2$$

realise the respective mappings. Here, it is important to recall that $p_i \geq 7$ and that 2 has a multiplicative inverse. However, the polynomials above are generally *not* realising the identity in $\mathbb{Z}/p_j\mathbb{Z}$ for $j \neq i$, which is required by the definition of flip_i, eqzero_i and eqone_i. For instance, in $\mathbb{Z}/7\mathbb{Z}$ we do not have $x^2 - 2x + 2 \equiv x$. Thus, for each of the three polynomials $p_{\mathit{flip}}(x)$, $p_{\mathit{eqzero}}(x)$, $p_{\mathit{eqone}}(x)$, written as $a_2x^2 + a_1x + a_0$, in order to obtain corresponding polynomials $p_{\mathit{flip},i}(x)$, $p_{\mathit{eqzero},i}(x)$, $p_{\mathit{eqone},i}(x)$, we apply the Chinese remainder theorem and for every $k \in \{0, 1, 2\}$ replace a_k with a'_k, where a'_k is the unique solution in $\mathbb{Z}/P\mathbb{Z}$ to the system of congruences $x \equiv a_k \bmod p_i$ and $x \equiv b_k \bmod p_j$ for each $1 \leq j \neq i \leq n$ with $b_1 \stackrel{\text{def}}{=} 1$ and $b_0 = b_2 \stackrel{\text{def}}{=} 0$. $\square$

$$(q,i,b) \xrightarrow{eqzero_{i+1}} (q,i+1,0) \text{ if } b=b', d=\rightarrow \qquad (q,i,b) \xrightarrow{eqzero_{i+1}\circ flip_i} (q,i+1,0) \text{ if } b\neq b', d=\rightarrow$$

$$(q,i,b) \xrightarrow{eqone_{i+1}} (q,i+1,1) \text{ if } b=b', d=\rightarrow \qquad (q,i,b) \xrightarrow{eqone_{i+1}\circ flip_i} (q,i+1,1) \text{ if } b\neq b', d=\rightarrow$$

$$(q,i,b) \xrightarrow{eqzero_{i-1}} (q,i-1,0) \text{ if } b=b', d=\leftarrow \qquad (q,i,b) \xrightarrow{eqzero_{i-1}\circ flip_i} (q,i-1,0) \text{ if } b\neq b', d=\leftarrow$$

$$(q,i,b) \xrightarrow{eqone_{i-1}} (q,i-1,1) \text{ if } b=b', d=\leftarrow \qquad (q,i,b) \xrightarrow{eqone_{i-1}\circ flip_i} (q,i-1,1) \text{ if } b\neq b', d=\leftarrow$$

Fig. 2. Transitions of $\mathcal{R}$ for simulating a transition (q,b,q',b',d) of $\mathcal{M}$.

We have now accumulated all ingredients that enable us to simulate $\mathcal{M}$ with a PRM $\mathcal{R}$. Subsequently, we will identify each mapping $flip_i$, $eqzero_i$ and $eqone_i$ with the corresponding polynomial from Lemma 2. We now define the control locations of Q, the transitions Δ and the labelling function λ of $\mathcal{R}$. The control locations of $\mathcal{R}$ contain those of $\mathcal{M}$ paired with the head position and a guess of the contents of the tape cell at the current head position, *i.e.*, $Q \stackrel{\text{def}}{=} Q_{\mathcal{M}} \times \{0,\ldots,n+1\} \times \{0,1\}$.

For every control location (q,i,b) of $\mathcal{R}$ such that $1 \leq i \leq n$ and every transition $(q,b,q',b',d) \in \Delta_{\mathcal{M}}$ of $\mathcal{M}$, Δ contains the transitions shown in Figure 2, and an additional transition $(q_f,i,0) \xrightarrow{x-P} (q_f,i,0)$ for each $b \in \{0,1\}$. The degree of the polynomials in Figure 2 is actually four, but quadratic polynomials can be regained by replacing a single transition with two consecutive transitions. Also, for brevity we have omitted the cases when the head moves to position 0 or $n+1$, whose behaviour can easily be hard-wired into $\mathcal{R}$.

The transitions of $\mathcal{R}$ are chosen such that every time we simulate a move of the head of $\mathcal{M}$, we guess the contents of the next tape cell. The guess is instantaneously verified through the application of the polynomials $eqzero_{i-1}$, $eqzero_{i+1}$, $eqone_{i-1}$ and $eqone_{i+1}$ along the transition: if the guess was wrong, the value of the register x becomes 2 modulo some prime p_i and will remain 2 modulo this prime forever. Simulating writing to a cell is done via the $flip_i$ polynomials, which are only applied if the currently read bit differs from the bit that is ought to be written. Finally, there is a self-loop at the control locations (q_f,i,b) subtracting P allows for checking that we end with a register value z such that $z \equiv 0 \bmod P$. Setting $q_{\mathcal{R}} = (q_0,0,0)$ and $q'_{\mathcal{R}} = (q_f,0,0)$, by induction on the length of the run of $\mathcal{M}$ and $\mathcal{R}$ respectively, it is easily verified that $(q_0, \triangleright 0^n \triangleleft, 0) \rightarrow^*_{\mathcal{M}} (q_f, \triangleright 0^n \triangleleft, 0)$ if, and only if, $q_{\mathcal{R}}(0) \rightarrow^*_{\mathcal{R}} q'_{\mathcal{R}}(0)$.

3.2 Membership in PSPACE

We now show the existence of a PSPACE algorithm that decides reachability in PRMs in the most unconstrained case where register values come from $\mathbb{Z}$. We will generalise this to the case of general formulas in the end of this section. Due to space constraints, it is not possible to give all technical details and formal proofs, we rather prefer presenting our algorithm on a high level and only state the most important technical results that give the PSPACE upper bound. All

formal details can be found in the appendix of the extended version of this paper.

For the remainder of this section, let us fix a PRM $\mathcal{R} = (Q, \Delta, \lambda, \phi)$ with one register x and control locations $q, q' \in Q$ for which we wish to decide $q(0) \rightarrow^*_{\mathcal{R}} q'(0)$. Denote by a and d the largest absolute value of all coefficients of the update polynomials in $\mathcal{R}$ and their maximum degree, respectively. We assume that every $p(x) \in P(\mathcal{R})$ is a non-constant polynomial. Otherwise, reachability can be reduced to a bounded number of reachability queries in PRMs with no constant update polynomials by guessing the order in which these transitions are traversed. The same approach would also enable us to additionally equip PRMs with zero tests.

Note that in the following, when referring to the size of a number, we refer to the number of bits required for its representation. On a high level, we can identify three key observations and ideas that lead us to our upper bound:

(i) There exists a bound b of size polynomial in $|\mathcal{R}|$ such that once the absolute register value of x goes above b, only counter-like polynomials can *decrement* the absolute value of x due to monotonicity properties of non-counter-like polynomials. A similar observation is part of the argument in [9] to show decidability of reachability for non-deterministic applications of affine polynomials.

(ii) The previous observation suggests that we should extract a 1-VASS $\mathcal{C}$ from the transitions from $\mathcal{R}$ labelled with counter-like polynomials that can simulate $\mathcal{R}$ acting on those transitions. This in turn enables us to make use of the property that reachability relations for 1-VASS are ultimately periodic with some period m of size polynomially bounded in $|\mathcal{C}|$ [10,11] and hence $|\mathcal{R}|$, and that reachability in 1-VASS can be decided in NP [12] and hence in PSPACE. In particular, this makes it possible to witness the existence of paths in $T(\mathcal{R})$ decrementing the register value from arbitrarily large absolute register values x, provided we know the residue class of x modulo m.

(iii) Observation (i) additionally enables us to show that paths in $T(\mathcal{R})$ whose absolute register value stays above b allow for deriving paths with special properties such that in particular residue classes modulo m of the register values occurring on the derived path are preserved. More precisely, we can derive paths for which a bound on the length of sequences that strictly decrease the absolute values of the register x exists. This in turn enables us to witness in PSPACE the *existence* of paths that end with a register value in a certain residue class modulo m by simulating $\mathcal{R}$ on residue classes modulo m without explicitly constructing those paths.

By gluing (ii) and (iii) together, we can then show that the PSPACE upper bound for reachability in PRMs follows. In the following, set $b \stackrel{\text{def}}{=} d(a+2)$. Observation (i) above is a consequence of the following lemma.

Lemma 3. *Let $p(x) \in P(\mathcal{R})$ be non-counter-like. Then $p(x)$ is monotonically increasing or decreasing in $\mathbb{Z} \setminus [-b, b]$, and $|p(z)| \geq 2|z|$ for all $z \in \mathbb{Z} \setminus [-b, b]$.*

The proof of the lemma is a straight-forward application of the inequality (1) in Section 2.1. It allows us to conclude that non-counter-like polynomials behave monotonically outside $[-b, b]$.

Before we start with formally discussing Observations (ii) and (iii), we need to introduce some auxiliary technical notation. A $q(z)$-$q'(z')$ *path* π *in* $T(\mathcal{R})$ *of length* n is a finite sequence of configurations $\pi : q_1(z_1)q_2(z_2)\cdots q_{n+1}(z_{n+1})$ such that $q(z) = q_1(z_1)$, $q'(z') = q_{n+1}(z_{n+1})$ and $q_i(z_i) \to_{\mathcal{R}} q_{i+1}(z_{i+1})$ for all $1 \leq i \leq n$. We write $\pi : q(z) \to^*_{\mathcal{R}} q'(z')$ if π is a $q(z)$-$q'(z')$ path and denote the length of π by $|\pi|$. Let $I \subseteq \mathbb{Z}$, we say that π *stays in* I if $z_i \in I$ for all $1 \leq i \leq n$. A path is *counter-like* if for all $q_i \xrightarrow{p(x)} q_{i+1}$, $p(x)$ is counter-like.

Now turning towards Observation (ii), the 1-VASS $\mathcal{C} \stackrel{\text{def}}{=} (Q_{\mathcal{C}}, \Delta_{\mathcal{C}}, \lambda_{\mathcal{C}})$ discussed above is obtained from the counter-like transitions of $\mathcal{R}$ as follows, where $\Delta_{\mathcal{C}} \stackrel{\text{def}}{=} \Delta_1 \cup \Delta_2$:

$$
\begin{aligned}
Q_{\mathcal{C}} &\stackrel{\text{def}}{=} \{q^{\sim} : q \in Q, \sim \in \{+,-\}\};\\
\Delta_1 &\stackrel{\text{def}}{=} \{(q_1^{\sim}, q_2^{\sim}) : q_1, q_2 \in Q, q_1 \xrightarrow{p(x)=x+a_0} q_2 \in \Delta\};\\
\Delta_2 &\stackrel{\text{def}}{=} \{(q_1^{\sim_1}, q_2^{\sim_2}) : q_1, q_2 \in Q, q_1 \xrightarrow{p(x)=-x+a_0} q_2 \in \Delta, \sim_1 \neq \sim_2\}\\
\lambda_{\mathcal{C}} &\stackrel{\text{def}}{=} (q_1^{\sim_1}, q_2^{\sim_2}) \mapsto x + \sim_2 a_0 \text{ if } q_1 \xrightarrow{a_1 x + a_0} q_2 \in \Delta.
\end{aligned}
$$

The idea behind this construction is as follows. The counter of $\mathcal{C}$ stores the *absolute value* of the register x of $\mathcal{R}$. The control locations of $\mathcal{C}$ are control locations from $\mathcal{R}$ with an indicator of the sign of the register x, *e.g.* q^- indicates that the control location is q and the value of the register x is negative. The transitions in Δ_1 and Δ_2 are defined such that they obey a flip of the sign. The following lemma, which can easily be shown by induction, enables us to relate paths in $T(\mathcal{R})$ and $T(\mathcal{C})$.

Lemma 4. *Let* $q_1(z_1), q_2(z_2) \in C(\mathcal{R})$ *and let* $z = \min\{|z_1|, |z_2|\}$ *such that* $z > a$. *There exists a counter-like path* $\pi : q_1(z_1) \to^*_{\mathcal{R}} q_2(z_2)$ *staying in* $\mathbb{Z}\backslash(-z, z)$ *if, and only if, there exists a path* $\pi' : q_1^{\sim_1}(|z_1| - z) \to^*_{\mathcal{C}} q_2^{\sim_2}(|z_2| - z)$ *for* $\sim_i = \operatorname{sgn}(z_i)$.

The benefit we get from extracting a 1-VASS from the counter-like transitions of $\mathcal{R}$ is that we can employ known periodicity properties for counter automata. The following proposition is a consequence of Lemma 5.1.9, pp. 139 in [11]. It allows us to conclude that reachability in 1-VASS is ultimately periodic with a small period of polynomial size.

Proposition 5 ([10,11]). *Let* $\mathcal{C} = (Q_{\mathcal{C}}, \Delta_{\mathcal{C}}, \lambda_{\mathcal{C}})$ *be a 1-VASS with maximum absolute increment* a. *There exists a fixed polynomial* p *and a period* $m \leq (|Q_{\mathcal{C}}|a)^{|Q_{\mathcal{C}}|}$ *such that for any* $q, q' \in Q_{\mathcal{C}}$ *and* $n' \in \mathbb{N}$ *there exists a set of residue classes* $R \subseteq \{0, \ldots, m-1\}$ *such that for all* $n > 2^{p(|\mathcal{C}|)} + n'$,

$$q(n) \to^*_{\mathcal{C}} q'(n') \text{ if, and only if, } n \equiv r \bmod m \text{ for some } r \in R.$$

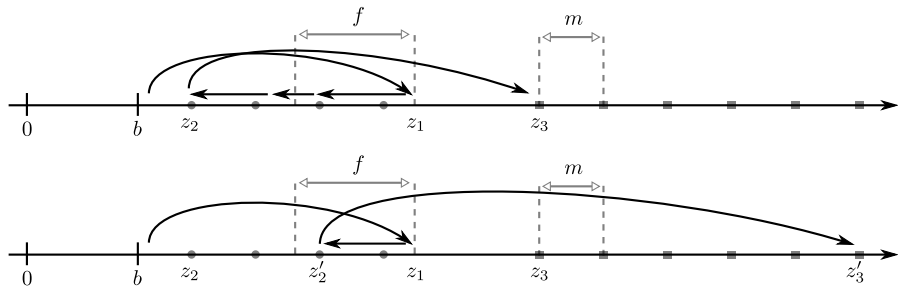

Fig. 3. Illustration of making a path non-dropping

For the remainder of this section, fix m to be the period from Proposition 5. We will now elaborate on Observation (iii) and turn towards normalising paths of $\mathcal{R}$ in $T(\mathcal{R})$ whose register values stay in $\mathbb{Z} \setminus [-b, b]$. To this end, we define a partial order $\preceq_m \subseteq \mathbb{Z} \times \mathbb{Z}$ such that

$$z_1 \preceq_m z_2 \text{ if } \operatorname{sgn}(z_1) = \operatorname{sgn}(z_2), |z_1| \leq |z_2| \text{ and } z_1 \equiv z_2 \bmod m.$$

Informally speaking, we show that if the register values encountered along a path fluctuate too much then we can decrease the magnitude of fluctuation while staying invariant w.r.t. $\preceq_m$. Formally, set $f \stackrel{\text{def}}{=} (2|Q|m)^2 + (2|Q|m)$ and let $\pi : q(z) \to^*_{\mathcal{R}} q'(z')$ be a path. We say that π is *dropping* if there are $1 \leq i < j < |\pi|$ such that $\pi(i) = q_i(z_i)$, $\pi(j) = q_j(z_j)$ and $|z_i| - |z_j| > f$. Observe that for any non-dropping path $\pi : q(z) \to q'(z')$, we have $|z'| \geq |z| - f$.

Lemma 6. *Let $\pi : q(z) \to^*_{\mathcal{R}} q'(z')$ be a path staying in $\mathbb{Z} \setminus [-b, b]$. If π is dropping then there is a path π' staying in $\mathbb{Z} \setminus [b, b]$ such that $\pi' : q(z) \to^*_{\mathcal{R}} q'(z'')$ for some $z'' \in \mathbb{Z}$, $z' \preceq_m z''$ and $|\pi'| < |\pi|$.*

Figure 3 illustrates the main idea. There, the illustrated path on top is dropping between register values z_1 and z_2. A counting argument shows that we can then find some z_2' in the interval $[z_1 - f, z_1]$ such that $z_2 \preceq_m z_2'$, which allows us to chop the path. We can then mimic the remainder of the path and end with some register value z_3' such that $z_3 \preceq_m z_3'$, illustrated at the bottom of Figure 3. A repeated application of the lemma allows us to make any path non-dropping, and it is not difficult to see that witnessing the existence of a non-dropping path reaching a certain residue class modulo m can be done in space polynomial in $|\mathcal{R}|$. This brings us to the main theorem of this paper.

Theorem 7. *Reachability in PRMs is* PSPACE*-complete.*

Proof (sketch). The main idea is that we can simulate $\mathcal{R}$ as long as its register values stay inside $[-B, B]$ for some sufficiently large $B \in \mathbb{N}$ of polynomial bit-size in $|\mathcal{R}|$. Here, it is important that checking whether an application of an update polynomial $p(x)$ to the current register value $z \in \mathbb{Z}$ leaves the interval $[-B, B]$

	counter polynomials			arbitrary polynomials		
	finite	upward closed	$\mathbb{Z}^d$	finite	upward closed	$\mathbb{Z}^d$
$d = 1$	PSPACE-c.[7]	NP-complete [12]		PSPACE-complete		
$d > 1$	PSPACE-c.[13]	EXPSPACE-h.,dec.[16,17]	NP-c.	PSPACE-c.	undec. [19]	

Table 1. Complexity landscape of reachability in d-PRMs

can be decided in polynomial time, and that $p(z)$ can be computed in polynomial time if $p(z) \in [-B, B]$ [5]. If the interval $[-B, B]$ were to be left, we compute $p(z) \bmod m$, guess a residue class r modulo m, and then check using Lemma 6 in polynomial space for the existence of a non-dropping path starting with register value $p(z) \bmod m$ reaching some register value in the residue class r. Moreover, we can use $\mathcal{C}$ together with Proposition 5 to check in polynomial space that from the residue class r there is a counter-like path back into $[-B, B]$. □

The proof of PSPACE-completeness can straight-forwardly be adapted to the case where the global invariant $\phi(x)$ imposes an upward-closed domain on the register x. The main difference is that $\mathcal{C}$ constructed above must not allow for flipping of signs, and when simulating $\mathcal{R}$ on the residue classes modulo m in the proof of Theorem 7 no transitions can be taken that result in a flip of the sign of the register.

Remark 8. A possible generalisation of PRMs could be to allow the global invariant to be a Presburger formula open in one variable x. Since the sets defined by such formulas are ultimately periodic below and above zero, it is not difficult to adapt the techniques used for showing the PSPACE upper bound in order to show that reachability is decidable. However, unsurprisingly the complexity of reachability may potentially increase by several exponents.

4 Concluding Remarks

This paper introduced polynomial register machines, a class of infinite-state systems comprising a finite number of integer-valued registers, whose domain is constrained by a linear constraint, with a finite-state controller which can update the registers along transitions by an application of a polynomial function. Our main result is that reachability with one register is PSPACE-complete. For higher dimensions, as discussed in the introduction, reachability becomes quickly undecidable, in particular already in the presence of two integer-valued registers and affine polynomials with *integer* coefficients [19].

A detailed complexity landscape classifying the complexity of reachability according to the number of registers, the type of update polynomials and the domain constraint is given in Table 1 together with bibliographic references. The results of this paper are emphasised by grey background colour.

References

1. D. Babić, B. Cook, A. J. Hu, and Z. Rakamarić. Proving termination of nonlinear command sequences. *Formal Aspects of Computing*, pages 1–15, 2012.
2. P. Bell and I. Potapov. On undecidability bounds for matrix decision problems. *Theoretical Computer Science*, 391(12):3–13, 2008.
3. R. Bonnet. The reachability problem for vector addition systems with one zero-test. In *Proc. MFCS*, volume 6907 of *LNCS*, pages 145–157, 2011.
4. A. R. Bradley, Z. Manna, and H. B. Sipma. Termination of polynomial programs. In *Proc. VMCAI*, volume 3385 of *LNCS*, pages 113–129. Springer, 2005.
5. F. Cucker, P. Koiran, and S. Smale. A polynomial time algorithm for Diophantine equations in one variable. *Journal of Symbolic Computation*, 27(1):21–29, 1999.
6. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset nets between decidability and undecidability. In *Proc. ICALP*, volume 1443 of *LNCS*, pages 103–115, 1998.
7. J. Fearnley and M. Jurdziński. Reachability in two-clock timed automata is PSPACE-complete. In *Proc. ICALP*, LNCS. Springer, 2013. To appear.
8. A. Finkel, P. McKenzie, and C. Picaronny. A well-structured framework for analysing Petri net extensions. *Information and Computation*, 195(1-2):1–29, 2004.
9. D. Fremont. The reachability problem for affine functions on the integers. Available online at: `http://web.mit.edu/~dfremont/www/reachability.pdf`, 2012.
10. S. Göller, C. Haase, J. Ouaknine, and J. Worrell. Branching-time model checking of parametric one-counter automata. In *Proc. FoSSaCS*, volume 7213 of *LNCS*, pages 406–420. Springer, 2012.
11. C. Haase. *On the Complexity of Model Checking Counter Automata*. PhD thesis, University of Oxford, UK, 2012.
12. C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *Proc. CONCUR*, volume 5710 of *LNCS*, pages 369–383. Springer, 2009.
13. C. Haase, J. Ouaknine, and J. Worrell. On the relationship between reachability problems in timed and counter automata. In *Proc. RP*, volume 7550 of *LNCS*, pages 54–65. Springer, 2012.
14. H. P. Hirst and W. T. Macey. Bounding the roots of polynomials. *The College Mathematics Journal*, 28(4):292–295, 1997.
15. J.-L. Lambert. A structure to decide reachability in petri nets. *Theoretical Computer Science*, 99(1):79–104, 1992.
16. R. Lipton. The reachability problem is exponential-space-hard. Technical report, Yale University, New Haven, CT, 1976.
17. E. W. Mayr. An algorithm for the general Petri net reachability problem. In *Proc. STOC*, pages 238–246, New York, NY, USA, 1981. ACM.
18. M. L. Minsky. Recursive Unsolvability of Post's Problem of "Tag" and other Topics in Theory of Turing Machines. *The Annals of Mathematics*, 74(3):437–455, 1961.
19. J. Reichert. Personal communication, 2013.

A Missing Proofs

A.1 Missing proofs from Section 3.2

Before proving the remaining lemmas, we recall and introduce some additional notation. Recall that a and d are the maximum absolute value of all coefficients and the maximum degree of all $p(x) \in P(\mathcal{R})$, respectively; $b = d(a+2)$; and m is as in Proposition 5.

Lemma 3. *Let $p(x) \in P(\mathcal{R})$ be non-counter-like. Then $p(x)$ is monotonically increasing or decreasing in $\mathbb{Z} \setminus [-b, b]$, and $|p(z)| \geq 2|z|$ for all $z \in \mathbb{Z} \setminus [-b, b]$.*

Proof. Write $p(x) = a_n x^n + \cdots + a_1 x + a_0$, where $1 \leq n \leq d$ and $|a_i| \leq a$. Regarding monotonicity, it suffices to show that the first derivative $p'(x)$ has no roots in I. We have $p'(x) = \sum_{i=0}^{n-1}(i+1)a_{i+1}x^i$. If $n = 1$ then $p'(x)$ has no roots at all. Otherwise for $n > 1$, from (1) we derive that for any root $c \in \mathbb{C}$ of $p'(x)$ we have

$$|c| \leq 1 + \sum_{i=0}^{n-2} |((i+1)a_{i+1})/(na_n)| \leq 1 + \sum_{i=0}^{n-2} a \leq da.$$

For showing $|p(z)| \geq 2|z|$, we wish to show that $p(x)$ has no intersection with $f_1(x) \stackrel{\text{def}}{=} 2x$ and $f_2(x) \stackrel{\text{def}}{=} -2x$ in $\mathbb{Z} \setminus [-b, b]$. Equating $p(x)$ with $f_1(x)$ and $f_2(x)$, we consequently have to bound the roots of $p_j(x) \stackrel{\text{def}}{=} p(x) + (-1)^j 2x$ for $j \in \{1, 2\}$. Let $c \in \mathbb{C}$ be any root of the $p_j(x)$. Applying (1), we get

$$\begin{aligned} |c| &\leq 1 + |a_0/a_n| + |(a_1 + (-1)^j 2)/a_n| + \sum_{i=2}^{n-1} |a_i/a_n| \\ &\leq d(a+2) \end{aligned}$$

Consequently, $|p(z)| \geq 2|z|$ for all $z \in \mathbb{Z} \setminus [-b, b]$. □

For $\pi : q_1(z_1)q_2(z_2)\cdots q_n(z_n)$, for $1 \leq i \leq n$ we denote by $\pi(i)$ the configuration $q_i(z_i)$. Given $\pi' : q'_1(z'_1)q'_2(z'_2)\cdots q'_m(z'_m)$ such that $q_n(z_n) \to_{\mathcal{R}} q'_1(z'_1)$, we denote by $\pi'' : \pi_1 \cdot \pi_2 \stackrel{\text{def}}{=} q_1(z_1)q_2(z_2)\cdots q_n(z_n)q'_1(z'_1)q'_2(z'_2)\cdots q'_m(z'_m)$ the concatenation of π_1 and π_2.

Lemma 4. *Let $q_1(z_1), q_2(z_2) \in C(\mathcal{R})$ and let $z = \min\{|z_1|, |z_2|\}$ such that $z > a$. There exists a counter-like path $\pi : q_1(z_1) \to^*_{\mathcal{R}} q_2(z_2)$ staying in $\mathbb{Z} \setminus (-z, z)$ if, and only if, there exists a path $\pi' : q_1^{\sim_1}(|z_1| - z) \to^*_{\mathcal{C}} q_2^{\sim_2}(|z_2| - z)$ for $\sim_i = \operatorname{sgn}(z_i)$.*

Proof. The proof is by induction on $n = |\pi|$. For the induction step, let $\pi = q_1(z_1)\cdots q_n(z_n)q_{n+1}(z_{n+1})$ be such that $q_n \xrightarrow{p(x)} q_{n+1}$ for some counter-like polynomial $p(x) = a_1 x + a_0$. By the induction hypothesis, there exists a path $\pi'_n : q_1^{\sim_1}(|z_1| - z) \to^*_{\mathcal{C}} q_n^{\sim_n}(|z_n| - z)$ for $z = \min\{|z_1|, |z_n|\}$ and $\sim_i = \operatorname{sgn}(z_i)$.

Hence $\pi'_n : q_1^{\sim 1}(|z_1| - z + k) \to^*_{\mathcal{C}} q_n^{\sim n}(|z_n| - z + k)$ for all $k \in \mathbb{N}$. If $a_1 < 0$ then $\operatorname{sgn}(z_{n+1}) = -\operatorname{sgn}(z_n)$, and if $a_1 > 0$ then $\operatorname{sgn}(z_{n+1}) = \operatorname{sgn}(z_n)$, since $a_0 \leq a$. Consequently, $|z_{n+1}| = |z_n| + a_1 \operatorname{sgn}(z_n) a_0$, and hence $\pi_n : q_1^{\sim 1}(|z_1| - z') \to^*_{\mathcal{C}} q_{n+1}^{\sim n+1}(|z_{n+1}| - z')$ for $z' = \min\{|z_1|, |z_n|, |z_{n+1}|\}$. The other direction follows analogously. □

Lemma 6. *Let $\pi : q(z) \to^*_{\mathcal{R}} q'(z')$ be a path staying in $\mathbb{Z} \setminus [-b, b]$. If π is dropping then there is a path π' staying in $\mathbb{Z} \setminus [-b, b]$ such that $\pi' : q(z) \to^*_{\mathcal{R}} q'(z'')$ for some $z'' \in \mathbb{Z}$, $z' \preceq_m z''$ and $|\pi'| < |\pi|$.*

The proof of this lemma relies on an intermediate lemma that we establish in the following. The next lemma shows that paths in $T(\mathcal{R})$ staying in $\mathbb{Z} \setminus [-b, b]$ are invariant with respect to translations modulo $\preceq_m$.

Lemma 9. *Let $\pi : q_1(z_1) \to^*_{\mathcal{R}} q_2(z_2)$ be a path staying in $\mathbb{Z} \setminus [-b, b]$ and $z'_1 \in \mathbb{Z}$ such that $z_1 \preceq_m z'_1$. There exists $z'_2 \in \mathbb{Z}$ and a path $\pi' : q_1(z'_1) \to^*_{\mathcal{R}} q_2(z'_2)$ staying in $\mathbb{Z} \setminus [-b, b]$ such that $z_2 \preceq_m z'_2$ and $|\pi| = |\pi'|$.*

Proof. The path π' is obtained by mimicking π. We prove the statement by induction on $n = |\pi|$. For the induction step, let $\pi = q_1(z_1) \cdots q_n(z_n) q_{n+1}(z_{n+1})$. The induction hypothesis yields the existence of a path $\pi'' : q_1(z'_1) \to^*_{\mathcal{R}} q_n(z'_n)$ with $z_n \preceq_m z'_n$. Let $q_n \xrightarrow{p(x)} q_{n+1}$ and $z'_{n+1} = p(z'_n)$, we have $z_{n+1} \equiv z'_{n+1} \bmod m$. Moreover, since $|z_n| > b$ and $p(x)$ is monotonically in- or decreasing by Lemma 3, we get $\operatorname{sgn}(z_{n+1}) = \operatorname{sgn}(z'_{n+1})$ and $|z_{n+1}| \leq |z'_{n+1}|$. By setting $\pi' \stackrel{\text{def}}{=} \pi'' \cdot q_{n+1}(z'_{n+1})$, we obtained the required path staying in $\mathbb{Z} \setminus (-b, b)$. □

We can now prove Lemma 6. Suppose there are configurations $q_i(z_i)$ and $q_j(z_j)$ such that $|z_i| - |z_j| > f = (2|Q|m)^2 + 2|Q|m$. Since $z_i, z_j \in \mathbb{Z} \setminus [-b, b]$, it follows from Lemma 3 that the absolute value of the register can only be decremented by counter-like transitions, and hence does not decrease by more than $a \leq m$ in one step. Thus, there are $k > 2|Q|m$ distinct configurations $\pi(i_g) = q_{i_g}(z_{i_g})$, $1 \leq g \leq k$ such that $i = i_1 < \ldots < i_k = j$, $|z_{i_g}| > |z_{i_{g+1}}|$ for all $1 \leq g \leq k$. A counting argument implies that we find i_s and i_t such that $q_{i_s} = q_{i_t}$ and $c_{i_t} \preceq_m c_{i_s}$. Let $\pi_t : q_{i_t}(z_{i_t}) \to^*_{\mathcal{R}} q'(z')$ be the suffix of π reaching $q'(z')$ from $q_{i_t}(z_{i_t})$. Lemma 9 yields the existence of a path $\pi_{i_s} : q_{i_s}(z_{i_s}) \to^*_{\mathcal{R}} q'(z'')$ for some $z'' \in \mathbb{Z}$ such that $z' \preceq_m z''$. Now let $\pi'' : q(z) \to^*_{\mathcal{R}} q_{i_s}(z_{i_s})$ be the prefix of π reaching $q_{i_s}(z_{i_s})$ from $q(z)$. We set $\pi' \stackrel{\text{def}}{=} \pi'' \cdot \pi_{i_s}$, which is the required $q(z)$-$q'(z'')$ path staying in $\mathbb{Z} \setminus [-b, b]$.

We now turn towards the proof of our main theorem.

Theorem 7. *Reachability in PRM is* PSPACE*-complete.*

Again, before proving this theorem we establish an intermediate lemma.

Lemma 10. *Let $\pi : q(z) \to^*_{\mathcal{R}} q'(z')$ be a path staying in $\mathbb{Z} \setminus [-b, b]$ such that $|z| > b + 2f$ and $|\pi| > f$. Then there exists a non-dropping path $\pi' : q(z) \to^*_{\mathcal{R}} q'(z'')$ staying in $\mathbb{Z} \setminus [-b, b]$ such that $z' \preceq_m z''$ or $z'' \preceq_m z'$ and $|\pi'| < |\pi|$.*

Proof. It suffices to show that we can obtain a path π' of length smaller than π if any of the conditions is not fulfilled.

If π is dropping then by Lemma 6 we can obtain π' such that $|\pi'| < |\pi|$. Thus, we can assume w.l.o.g. that π is not dropping. Since $|\pi| > f$, as in Lemma 6, we find $1 \leq i < j \leq |\pi|$ such that $\pi(i) = q_i(z_i)$, $\pi(j) = q_j(z_j)$, $q_i = q_j$, and $z_i \preceq_m z_j$ or $z_j \preceq_m z_i$. Denote by $\pi_i : q(z) \rightarrow^*_{\mathcal{R}} q_i(z_i)$ the induced prefix and $\pi_j : q_j(z_j) \rightarrow^*_{\mathcal{R}} q'(z')$ the induced suffix of π. Since π is not dropping, both π_i and π_j are not dropping.

If $z_j \preceq_m z_i$ then we can apply Lemma 9, in order to obtain a path π' with the desired properties.

Otherwise, let $z_i \preceq_m z_j$. We have that $|z_i| > b + f$, and we can mimic π_j starting from $q_i(z_i)$ as in Lemma 9 while staying invariant w.r.t. $\preceq_m$. Here, it is crucial that π_j is not dropping. We have that π' does not drop by more than $2f$, hence it is a valid path staying in in $\mathbb{Z} \setminus [-b, b]$, from which we can again by application of Lemma 9 obtain the desired non-dropping path. □

We can now prove our main theorem. Set $I \stackrel{\text{def}}{=} [-B, B]$ for $B \stackrel{\text{def}}{=} b + m + 2^{p(|\mathcal{C}|)} + a$, where $p(|\mathcal{C}|)$ is as in Proposition 5. Suppose there is $\pi : q(0) \rightarrow^*_{\mathcal{R}} q'(0)$. W.l.o.g. we may assume that no configuration appears twice along π. We show that we can witness the existence of π using polynomial space only. If π stays inside I then $|\pi| \leq 2B$, and hence π can be guessed using polynomial space only. It is important to remark that guessing such a path π is actually possible, since a single application of a polynomial $p(x) \in P(\mathcal{R})$ can lead to a register value whose representation requires exponentially many bits due to the sparse encoding of polynomials. However, as shown in [5], deciding $|p(z)| > B$ can be performed in polynomial time, and under the assumption that $p(z) \in I$ the value of $p(z)$ can also be computed in polynomial time.

Otherwise, if π does not stay in I, we can decompose it as

$$\pi = \pi_1 \cdot \gamma_1 \cdot q_1(z_1) \cdot \pi_2 \cdot \gamma_2 \cdot q_2(z_2) \cdots \pi_n \cdot \gamma_n \cdot q_n(z_n) \cdot \pi_{n+1}$$

such that all π_i stay inside I, all γ_i stay inside $\mathbb{Z} \setminus I$, and all $|z_i| \in [B, B + a]$. Obviously, n is bounded by $2a$. As before, all π_i can be guessed in polynomial space, so it remains to show that we can prove in polynomial space the existence of the γ_i. Set $D \stackrel{\text{def}}{=} B + a + 2^{p(\mathcal{C})} + m + 2f$. For a fixed γ_i, if γ_i stays in $(-D, D)$ then it can be guessed in polynomial space as before. Otherwise, γ_i can be decomposed as

$$\gamma_i = \gamma_i' \cdot q_i'(z_i') \cdot \gamma_i''$$

such that γ_i' stays in $(-D, D)$ and $|z_i'| > D$. As before, γ_i' can non-deterministically be guessed. So it remains to show that the existence of a path $q_i'(z_i') \rightarrow^*_{\mathcal{R}} q_i(z_i)$ can be witnessed in polynomial space. Since the absolute value of the register x can only be decreased by at most a along γ_i'', we find some j such that $0 \leq j \leq |\gamma_i''|$ and $\gamma_i''(j) = s(y)$ such that $|y| \in [B+a+2^{p(\mathcal{C})}, B+a+2^{p(\mathcal{C})}+m]$ and γ_i'' is counter-like from position j onwards. From Proposition 5, we can conclude

that $s(y + \mathrm{sgn}(y)km) \to^*_{\mathcal{R}} q_i(z_1)$ for all $k \geq 0$. On the other hand, Lemma 10 allows us to decide the existence of a non-dropping path $q'_i(z'_i) \to^*_{\mathcal{R}} s(y')$ for some y' such that $y \preceq_m y'$ in polynomial space. Hence, $q'_i(z'_i) \to^*_{\mathcal{R}} q_i(z_i)$ can be witnessed by guessing $s(y)$, checking if there is a counter-like path $s(y) \to^*_{\mathcal{R}} q_i(z_i)$, and checking for the existence of a non-dropping path $q'_i(z'_i) \to^*_{\mathcal{R}} s(y')$ for some y' such that $y \preceq_m y'$, all of which can be performed in polynomial space.