Stability and Complexity of Minimising Probabilistic Automata[*]

Stefan Kiefer and Björn Wachter

University of Oxford, UK

Abstract. We consider the state-minimisation problem for weighted and probabilistic automata. We provide a numerically stable polynomial-time minimisation algorithm for weighted automata, with guaranteed bounds on the numerical error when run with floating-point arithmetic. Our algorithm can also be used for "lossy" minimisation with bounded error. We show an application in image compression. In the second part of the paper we study the complexity of the minimisation problem for probabilistic automata. We prove that the problem is NP-hard and in PSPACE, improving a recent EXPTIME-result.

1 Introduction

Probabilistic and weighted automata were introduced in the 1960s, with many fundamental results established by Schützenberger [25] and Rabin [23]. Nowadays probabilistic automata are widely used in automated verification, natural-language processing, and machine learning.

Probabilistic automata (PAs) generalise deterministic finite automata (DFAs): The transition relation specifies, for each state q and each input letter a, a probability distribution on the successor state. Instead of a single initial state, a PA has a probability distribution over states; and instead of accepting states, a PA has an acceptance probability for each state. As a consequence, the language induced by a PA is a *probabilistic language*, i.e., a mapping $L : \Sigma^* \to [0,1]$, which assigns each word an acceptance probability. Weighted automata (WAs), in turn, generalise PAs: the numbers appearing in the specification of a WA may be arbitrary real numbers. As a consequence, a WA induces a *weighted language*, i.e., a mapping $L : \Sigma^* \to \mathbb{R}$. Loosely speaking, the weight of a word w is the sum of the weights of all accepting w-labelled paths through the WA.

Given an automaton, it is natural to ask for a small automaton that accepts the same weighted language. A small automaton is particularly desirable when further algorithms are run on the automaton, and the runtime of those algorithms depends crucially on the size of the automaton [17]. In this paper we consider the problem of minimising the number of states of a given WA or PA, while preserving its (weighted or probabilistic) language.

WAs can be minimised in polynomial time, using, e.g., the standardisation procedure of [25]. When implemented efficiently (for instance using triangular

---

[*] For a full version of this paper, see [19].

matrices), one obtains an $O(|\Sigma|n^3)$ minimisation algorithm, where n is the number of states. As PAs are special WAs, the same holds in principle for PAs.

There are two problems with these algorithms: (1) numerical instability, i.e., round-off errors can lead to an automaton that is not minimal and/or induces a different probabilistic language; and (2) minimising a PA using WA minimisation algorithms does not necessarily result in a PA: transition weights may, e.g., become negative. This paper deals with those two issues.

Concerning problem (1), numerical stability is crucial under two scenarios: (a) when the automaton size makes the use of exact rational arithmetic prohibitive, and thus necessitates floating-point arithmetic [17]; or (b) when exact minimisation yields an automaton that is still too large and a "lossy compression" is called for, as in image compression [15]. Besides finding a numerically stable algorithm, we aim at two further goals: First, a stable algorithm should also be efficient; i.e., it should be as fast as classical (efficient, but possibly unstable) algorithms. Second, stability should be provable, and ideally there should be easily computable error bounds. In Section 3 we provide a numerically stable $O(|\Sigma|n^3)$ algorithm for minimising WAs. The algorithm generalises the *Arnoldi iteration* [2] which is used for locating eigenvalues in numerical linear algebra. The key ingredient, leading to numerical stability and allowing us to give error bounds, is the use of special orthonormal matrices, called *Householder reflectors* [14]. To the best of the authors' knowledge, these techniques have not been previously utilised for computations on weighted automata.

Problem (2) suggests a study of the computational complexity of the *PA minimisation problem*: given a PA and $m \in \mathbb{N}$, is there an equivalent PA with m states? In the 1960s and 70s, PAs were studied extensively, see the survey [7] for references and Paz's influential textbook [22]. PAs appear in various flavours and under different names. For instance, in *stochastic sequential machines* [22] there is no fixed initial state distribution, so the semantics of a stochastic sequential machine is not a probabilistic language, but a mapping from initial distributions to probabilistic languages. This gives rise to several notions of minimality in this model [22]. In this paper we consider only PAs with an initial state distribution; equivalence means equality of probabilistic languages.

One may be tempted to think that PA minimisation is trivially in NP, by guessing the minimal PA and verifying equivalence. However, it is not clear that the minimal PA has rational transition probabilities, even if this holds for the original PA.

For DFAs, which are special PAs, an automaton is minimal (i.e., has the least number of states) if and only if all states are reachable and no two states are equivalent. However, this equivalence does in general not hold for PAs. In fact, even if a PA has the property that no state behaves like a convex combination of other states, the PA may nevertheless not be minimal. As an example, consider the PA in the middle of Figure 2 on page 276. State 3 behaves like a convex combination of states 2 and 4: state 3 can be removed by splitting its incoming arc with weight 1 in two arcs with weight 1/2 each and redirecting the new arcs to states 2 and 4. The resulting PA is equivalent and no state can be replaced

by a convex combination of other states. But the PA on the right of the figure is equivalent and has even fewer states.

In Section 4 we show that the PA minimisation problem is NP-hard by a reduction from 3SAT. A step in our reduction is to show that the following problem, the *hypercube problem*, is NP-hard: given a convex polytope P within the d-dimensional unit hypercube and $m \in \mathbb{N}$, is there a convex polytope with m vertices that is nested between P and the hypercube? We then reduce the hypercube problem to PA minimisation. To the best of the authors' knowledge, no lower complexity bound for PA minimisation has been previously obtained, and there was no reduction from the hypercube problem to PA minimisation. However, towards the converse direction, the textbook [22] suggests that an algorithm for the hypercube problem could serve as a "subroutine" for a PA minimisation algorithm, leaving the decidability of both problems open. In fact, problems similar to the hypercube problem were subsequently studied in the field of computational geometry, citing PA minimisation as a motivation [26,21,11,10].

The PA minimisation problem was shown to be decidable in [20], where the authors provided an exponential reduction to the existential theory of the reals, which, in turn, is decidable in PSPACE [8,24], but not known to be PSPACE-hard. In Section 4.2 we give a polynomial-time reduction from the PA minimisation problem to the existential theory of the reals. It follows that the PA minimisation problem is in PSPACE, improving the EXPTIME result of [20].

2   Preliminaries

In the technical development that follows it is more convenient to talk about vectors and transition matrices than about states, edges, alphabet labels and weights. However, a PA "of size n" can be easily viewed as a PA with states $1, 2, \ldots, n$. We use this equivalence in pictures.

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. For $n \in \mathbb{N}$ we write $\mathbb{N}_n$ for the set $\{1, 2, \ldots, n\}$. For $m, n \in \mathbb{N}$, elements of $\mathbb{R}^m$ and $\mathbb{R}^{m \times n}$ are viewed as vectors and matrices, respectively. Vectors are row vectors by default. Let $\alpha \in \mathbb{R}^m$ and $M \in \mathbb{R}^{m \times n}$. We denote the entries by $\alpha[i]$ and $M[i, j]$ for $i \in \mathbb{N}_m$ and $j \in \mathbb{N}_n$. By $M[i, \cdot]$ we refer to the ith row of M. By $\alpha[i..j]$ for $i \leq j$ we refer to the sub-vector $(\alpha[i], \alpha[i + 1], \ldots, \alpha[j])$, and similarly for matrices. We denote the transpose by α^T (a column vector) and $M^T \in \mathbb{R}^{n \times m}$. We write I_n for the $n \times n$ identity matrix. When the dimension is clear from the context, we write $e(i)$ for the vector with $e(i)[i] = 1$ and $e(i)[j] = 0$ for $j \neq i$. A vector $\alpha \in \mathbb{R}^m$ is *stochastic* if $\alpha[i] \geq 0$ for all $i \in \mathbb{N}_m$ and $\sum_{i=1}^m \alpha[i] \leq 1$. A matrix is *stochastic* if all its rows are stochastic. By $\|\cdot\| = \|\cdot\|_2$, we mean the 2-norm for vectors and matrices throughout the paper unless specified otherwise. If a matrix M is stochastic, then $\|M\| \leq \|M\|_1 \leq 1$. For a set $V \subseteq \mathbb{R}^n$, we write $\langle V \rangle$ to denote the vector space spanned by V, where we often omit the braces when denoting V. For instance, if $\alpha, \beta \in \mathbb{R}^n$, then $\langle \{\alpha, \beta\} \rangle = \langle \alpha, \beta \rangle = \{r\alpha + s\beta \mid r, s \in \mathbb{R}\}$.

An $\mathbb{R}$-*weighted automaton (WA)* $\mathcal{A} = (n, \Sigma, M, \alpha, \eta)$ consists of a size $n \in \mathbb{N}$, a finite alphabet Σ, a map $M : \Sigma \to \mathbb{R}^{n \times n}$, an initial (row) vector

$\alpha \in \mathbb{R}^n$, and a final (column) vector $\eta \in \mathbb{R}^n$. Extend M to Σ^* by setting $M(a_1 \cdots a_k) := M(a_1) \cdots M(a_k)$. The *language* $L_{\mathcal{A}}$ of a WA $\mathcal{A}$ is the mapping $L_{\mathcal{A}} : \Sigma^* \to \mathbb{R}$ with $L_{\mathcal{A}}(w) = \alpha M(w) \eta$. WAs $\mathcal{A}, \mathcal{B}$ over the same alphabet Σ are said to be *equivalent* if $L_{\mathcal{A}} = L_{\mathcal{B}}$. A WA $\mathcal{A}$ is *minimal* if there is no equivalent WA $\mathcal{B}$ of smaller size.

A *probabilistic automaton (PA)* $\mathcal{A} = (n, \Sigma, M, \alpha, \eta)$ is a WA, where α is stochastic, $M(a)$ is stochastic for all $a \in \Sigma$, and $\eta \in [0, 1]^n$. A PA is a *DFA* if all numbers in M, α, η are 0 or 1.

3   Stable WA Minimisation

In this section we discuss WA minimisation. In Section 3.1 we describe a WA minimisation algorithm in terms of elementary linear algebra. The presentation reminds of Brzozowski's algorithm for NFA minimisation [6].[1] WA minimisation techniques are well known, originating in [25], cf. also [4, Chapter II] and [3]. Our algorithm and its correctness proof may be of independent interest, as they appear to be particularly succinct. In Sections 3.2 and 3.3 we take further advantage of the linear algebra setting and develop a numerically stable WA minimisation algorithm.

3.1   Brzozowski-like WA Minimisation

Let $\mathcal{A} = (n, \Sigma, M, \alpha, \eta)$ be a WA. Define the *forward space* of $\mathcal{A}$ as the (row) vector space $\mathsf{F} := \langle \alpha M(w) \mid w \in \Sigma^* \rangle$. Similarly, let the *backward space* of $\mathcal{A}$ be the (column) vector space $\mathsf{B} := \langle M(w) \eta \mid w \in \Sigma^* \rangle$. Let $\overrightarrow{n} \in \mathbb{N}$ and $F \in \mathbb{R}^{\overrightarrow{n} \times n}$ such that the rows of F form a basis of F. Similarly, let $\overleftarrow{n} \in \mathbb{N}$ and $B \in \mathbb{R}^{n \times \overleftarrow{n}}$ such that the columns of B form a basis of B. Since $\mathsf{F}M(a) \subseteq \mathsf{F}$ and $M(a)\mathsf{B} \subseteq \mathsf{B}$ for all $a \in \Sigma$, there exist maps $\overrightarrow{M} : \Sigma \to \mathbb{R}^{\overrightarrow{n} \times \overrightarrow{n}}$ and $\overleftarrow{M} : \Sigma \to \mathbb{R}^{\overleftarrow{n} \times \overleftarrow{n}}$ such that

$$FM(a) = \overrightarrow{M}(a)F \quad \text{and} \quad M(a)B = B\overleftarrow{M}(a) \quad \text{for all } a \in \Sigma. \tag{1}$$

We call $(F, \overrightarrow{M})$ a *forward reduction* and $(B, \overleftarrow{M})$ a *backward reduction*. We will show that minimisation reduces to computing such reductions. By symmetry we can focus on forward reductions. We call a forward reduction $(F, \overrightarrow{M})$ *canonical* if $F[1, \cdot]$ (i.e., the first row of F) is a multiple of α, and the rows of F are orthonormal, i.e., $FF^T = I_{\overrightarrow{n}}$.

Let $\mathcal{A} = (n, \Sigma, M, \alpha, \eta)$ be a WA with forward and backward reductions $(F, \overrightarrow{M})$ and $(B, \overleftarrow{M})$, respectively. Let $\overrightarrow{\alpha} \in \mathbb{R}^{\overrightarrow{n}}$ be a row vector such that $\alpha = \overrightarrow{\alpha} F$; let $\overleftarrow{\eta} \in \mathbb{R}^{\overleftarrow{n}}$ be a column vector such that $\eta = B\overleftarrow{\eta}$. (If $(F, \overrightarrow{M})$ is canonical, we have $\overrightarrow{\alpha} = (\pm \|\alpha\|, 0, \ldots, 0)$.) Call $\overrightarrow{\mathcal{A}} := (\overrightarrow{n}, \Sigma, \overrightarrow{M}, \overrightarrow{\alpha}, F\eta)$ a *forward WA* of $\mathcal{A}$ with base F and $\overleftarrow{\mathcal{A}} := (\overleftarrow{n}, \Sigma, \overleftarrow{M}, \alpha B, \overleftarrow{\eta})$ a *backward WA* of $\mathcal{A}$ with base B. By extending (1) one can see that these automata are equivalent to $\mathcal{A}$:

---

[1] In [5] a very general Brzozowski-like minimization algorithm is presented in terms of universal algebra. One can show that it specialises to ours in the WA setting.

Proposition 1. *Let $\mathcal{A}$ be a WA. Then $L_{\mathcal{A}} = L_{\overrightarrow{\mathcal{A}}} = L_{\overleftarrow{\mathcal{A}}}$.*

Further, applying both constructions consecutively yields a minimal WA:

Theorem 2. *Let $\mathcal{A}$ be a WA. Let $\mathcal{A}' = \overleftrightarrow{\mathcal{A}}$ or $\mathcal{A}' = \overrightarrow{\overleftarrow{\mathcal{A}}}$. Then $\mathcal{A}'$ is minimal and equivalent to $\mathcal{A}$.*

Theorem 2 mirrors Brzozowski's NFA minimisation algorithm. We give a short proof in [19].

3.2 Numerically Stable WA Minimisation

Theorem 2 reduces the problem of minimising a WA to the problem of computing a forward and a backward reduction. In the following we focus on computing a *canonical* (see above for the definition) forward reduction $(F, \overrightarrow{M})$. Figure 1 shows a generalisation of Arnoldi's iteration [2] to multiple matrices. Arnoldi's iteration is typically used for locating eigenvalues [12]. Its generalisation to multiple matrices is novel, to the best of the authors's knowledge. Using (1) one can see that it computes a canonical forward reduction by iteratively extending a partial orthonormal basis $\{f_1, \ldots, f_j\}$ for the forward space F.

function ArnoldiReduction
input: $\alpha \in \mathbb{R}^n$; $M : \Sigma \to \mathbb{R}^{n \times n}$
output: canonical forward reduction $(F, \overrightarrow{M})$ with $F \in \mathbb{R}^{\overrightarrow{n} \times n}$ and $\overrightarrow{M} : \Sigma \to \mathbb{R}^{\overrightarrow{n} \times \overrightarrow{n}}$
  $\ell := 0$; $j := 1$; $f_1 := \alpha/\|\alpha\|$ (or $f_1 := -\alpha/\|\alpha\|$)
  while $\ell < j$ do
    $\ell := \ell + 1$
    for $a \in \Sigma$ do
      if $f_\ell M(a) \notin \langle f_1, \ldots, f_j \rangle$
        $j := j + 1$
        define f_j orthonormal to $f_1, \ldots, f_{j-1}$ such that
          $\langle f_1, \ldots, f_{j-1}, f_\ell M(a) \rangle = \langle f_1, \ldots, f_j \rangle$
        define $\overrightarrow{M}(a)[\ell, \cdot]$ such that $f_\ell M(a) = \sum_{i=1}^{j} \overrightarrow{M}(a)[\ell, i] f_i$
          and $\overrightarrow{M}(a)[\ell, j+1..n] = (0, \ldots, 0)$
  $\overrightarrow{n} := j$; form $F \in \mathbb{R}^{\overrightarrow{n} \times \overrightarrow{n}}$ with rows $f_1, \ldots, f_{\overrightarrow{n}}$
  return F and $\overrightarrow{M}(a)[1..\overrightarrow{n}, 1..\overrightarrow{n}]$ for all $a \in \Sigma$

Fig. 1. Generalised Arnoldi iteration

For efficiency, one would like to run generalised Arnoldi iteration (Figure 1) using floating-point arithmetic. This leads to *round-off errors*. The check "if $f_\ell M(a) \notin \langle f_1, \ldots, f_j \rangle$" is particularly problematic: since the vectors $f_1, \ldots, f_j$ are computed with floating-point arithmetic, we cannot expect that $f_\ell M(a)$ lies *exactly* in the vector space spanned by those vectors, even if that would be the case without round-off errors. As a consequence, we need to introduce an *error tolerance parameter* $\tau > 0$, so that the check "$f_\ell M(a) \notin \langle f_1, \ldots, f_j \rangle$" returns *true* only if $f_\ell M(a)$ has a "distance" of more than τ to the vector space

$\langle f_1, \ldots, f_j \rangle$.[2] Without such a "fuzzy" comparison the resulting automaton could even have more states than the original one. The error tolerance parameter τ causes further errors.

To assess the impact of those errors, we use the *standard model of floating-point arithmetic*, which assumes that the elementary operations $+, -, \cdot, /$ are computed exactly, up to a relative error of at most the *machine epsilon* $\varepsilon_{\text{mach}} \geq 0$. It is stated in [13, Chapter 2]: "This model is valid for most computers, and, in particular, holds for IEEE standard arithmetic." The bit length of numbers arising in a numerical computation is bounded by hardware, using suitable roundoff. So we adopt the convention of numerical linear algebra to take the number of arithmetic operations as a measure of time complexity.

The algorithm ArnoldiReduction (Figure 1) leaves open how to implement the conditional "if $f_\ell M(a) \notin \langle f_1, \ldots, f_j \rangle$", and how to compute the new basis element f_j. In [19] we propose an instantiation *HouseholderReduction* of Arnoldi-Reduction based on so-called *Householder reflectors* [14], which are special orthonormal matrices. We prove the following stability property:

Proposition 3. *Consider the algorithm* HouseholderReduction *in [19], which has the following interface:*

function HouseholderReduction
input: $\alpha \in \mathbb{R}^n$; $M : \Sigma \to \mathbb{R}^{n \times n}$; error tolerance parameter $\tau \geq 0$
output: canonical forward reduction $(F, \overrightarrow{M})$ with $F \in \mathbb{R}^{\overrightarrow{n} \times n}$ and $\overrightarrow{M} : \Sigma \to \mathbb{R}^{\overrightarrow{n} \times \overrightarrow{n}}$

We have:

1. *The number of arithmetic operations is $O(|\Sigma| n^3)$.*
2. *HouseholderReduction instantiates ArnoldiReduction.*
3. *The computed matrices satisfy the following error bound: For each $a \in \Sigma$, the matrix $\mathcal{E}(a) \in \mathbb{R}^{\overrightarrow{n} \times n}$ with $\mathcal{E}(a) := F M(a) - \overrightarrow{M}(a) F$ satisfies*

$$\|\mathcal{E}(a)\| \leq 2\sqrt{n}\tau + cmn^3 \varepsilon_{mach} \,,$$

   where $m > 0$ is such that $\|M(a)\| \leq m$ holds for all $a \in \Sigma$, and $c > 0$ is an input-independent constant.

The proof follows classical error-analysis techniques for QR factorisations with Householder reflectors [13, Chapter 19], but is substantially complicated by the presence of the "if" conditional and the resulting need for the τ parameter. By Proposition 3.2. HouseholderReduction computes a precise canonical forward reduction for $\varepsilon_{\text{mach}} = \tau = 0$. For positive $\varepsilon_{\text{mach}}$ and τ the error bound grows linearly in $\varepsilon_{\text{mach}}$ and τ, and with modest polynomials in the WA size n. In practice $\varepsilon_{\text{mach}}$ is very small[3], so that the term $cmn^3\varepsilon_{\text{mach}}$ can virtually be ignored.

The use of Householder reflectors is crucial to obtain the bound of Proposition 3. Let us mention a few alternative techniques, which have been used for computing certain matrix factorisations. Such factorisations (QR or LU) are

---

[2] This will be made formal in our algorithm.
[3] With IEEE double precision, e.g., it holds $\varepsilon_{\text{mach}} = 2^{-53}$ [13].

related to our algorithm. *Gaussian elimination* can also be used for WA minimisation in time $O(|\Sigma|n^3)$, but its stability is governed by the *growth factor*, which can be exponential even with *pivoting* [13, Chapter 9], so the bound on $\|\mathcal{E}(a)\|$ in Proposition 3 would include a term of the form $2^n \varepsilon_{\mathrm{mach}}$. The most straightforward implementation of ArnoldiReduction would use the *Classical Gram-Schmidt* process, which is highly unstable [13, Chapter 19.8]. A variant, the *Modified Gram-Schmidt* process is stable, but the error analysis is complicated by a possibly loss of orthogonality of the computed matrix F. The extent of that loss depends on certain condition numbers (cf. [13, Equation (19.30)]), which are hard to estimate or control in our case. In contrast, our error bound is independent of condition numbers.

Using Theorem 2 we can prove:

Theorem 4. *Consider the following algorithm:*

function HouseholderMinimisation
input: WA $\mathcal{A} = (n, \Sigma, M, \alpha, \eta)$; error tolerance parameter $\tau \geq 0$
output: minimised WA $\mathcal{A}' = (n', \Sigma, M', \alpha', \eta')$.
    compute forward reduction $(F, \overrightarrow{M})$ of $\mathcal{A}$ using HouseholderReduction
    form $\overrightarrow{\mathcal{A}} := (\overrightarrow{n}, \Sigma, \overrightarrow{M}, \overrightarrow{\alpha}, \overrightarrow{\eta})$ as the forward WA of $\mathcal{A}$ with base F
    compute backward reduction (B, M') of $\overrightarrow{\mathcal{A}}$ using HouseholderReduction
    form $\mathcal{A}' := (n', \Sigma, M', \alpha', \eta')$ as the backward WA of $\overrightarrow{\mathcal{A}}$ with base B
    return $\mathcal{A}'$

We have:

1. *The number of arithmetic operations is $O(|\Sigma|n^3)$.*
2. *For $\varepsilon_{mach} = \tau = 0$, the computed WA $\mathcal{A}'$ is minimal and equivalent to $\mathcal{A}$.*
3. *Let $\tau > 0$. Let $m > 0$ such that $\|A\| \leq m$ holds for all $A \in \{M(a), \overrightarrow{M}(a), M'(a) \mid a \in \Sigma\}$. Then for all $w \in \Sigma^*$ we have*

$$|L_{\mathcal{A}}(w) - L_{\mathcal{A}'}(w)| \leq 4|w|\|\alpha\|m^{|w|-1}\|\eta\|\sqrt{n}\tau$$
$$+ c\max\{|w|, 1\}\|\alpha\|m^{|w|}\|\eta\|n^3 \varepsilon_{mach} ,$$

where $c > 0$ is an input-independent constant.

The algorithm computes a backward reduction by running the straightforward backward variant of HouseholderReduction. We remark that for PAs one can take $m = 1$ for the norm bound m from part 3. of the theorem (or $m = 1 + \varepsilon$ for a small ε if unfortunate roundoff errors occur). It is hard to avoid an error bound exponential in the word length $|w|$, as $|L_{\mathcal{A}}(w)|$ itself may be exponential in $|w|$ (consider a WA of size 1 with $M(a) = 2$). Theorem 4 is proved in [19].

The error bounds in Proposition 3 and Theorem 4 suggest to choose a small value for the error tolerance parameter τ. But as we have discussed, the computed WA may be non-minimal if τ is set too small or even to 0, intuitively because round-off errors may cause the algorithm to overlook minimisation opportunities. So it seems advisable to choose τ smaller (by a few orders of magnitude) than the desired bound on $\|\mathcal{E}(a)\|$, but larger (by a few orders of magnitude) than $\varepsilon_{\mathrm{mach}}$.

Note that for $\varepsilon_{\mathrm{mach}} > 0$ Theorem 4 does not provide a bound on the number of states of $\mathcal{A}'$.

To illustrate the stability issue we have experimented with minimising a PA $\mathcal{A}$ derived from Herman's protocol as in [17]. The PA has 190 states and $\Sigma = \{a\}$. When minimising with the (unstable) Classical Gram-Schmidt process, we have measured a huge error of $|L_{\mathcal{A}}(a^{190}) - L_{\mathcal{A}'}(a^{190})| \approx 10^{36}$. With the Modified Gram-Schmidt process and the method from Theorem 4 the corresponding errors were about 10^{-7}, which is in the same order as the error tolerance parameter τ.

3.3 Lossy WA Minimisation

A larger error tolerance parameter τ leads to more "aggressive" minimisation of a possibly already minimal WA. The price to pay is a shift in the language: one would expect only $L'_{\mathcal{A}}(w) \approx L_{\mathcal{A}}(w)$. Theorem 4 provides a bound on this imprecision. In this section we illustrate the trade-off between size and precision using an application in image compression.

Weighted automata can be used for image compression, as suggested by Culik et al. [15]. An image, represented as a two-dimensional matrix of grey-scale values, can be encoded as a weighted automaton where each pixel is addressed by a unique word. To obtain this automaton, the image is recursively subdivided into quadrants. There is a state for each quadrant and transitions from a quadrant to its sub-quadrants. At the level of the pixels, the automaton accepts with the correct grey-scale value.

Following this idea, we have implemented a prototype tool for image compression based on the algorithm of Theorem 4. We give details and show example pictures in [19]. This application illustrates lossy minimisation. The point is that Theorem 4 guarantees bounds on the loss.

4   The Complexity of PA Minimisation

Given a PA $\mathcal{A} = (n, \Sigma, M, \alpha, \eta)$ and $n' \in \mathbb{N}$, the *PA minimisation problem* asks whether there exists a PA $\mathcal{A}' = (n', \Sigma, M', \alpha', \eta')$ so that $\mathcal{A}$ and $\mathcal{A}'$ are equivalent. For the complexity results in this section we assume that the numbers in the description of the given PA are fractions of natural numbers represented in binary, so they are rational. In Section 4.1 we show that the minimisation problem is NP-hard. In Section 4.2 we show that the problem is in PSPACE by providing a polynomial-time reduction to the existential theory of the reals.

4.1   NP-Hardness

We will show:

Theorem 5. *The PA minimisation problem is NP-hard.*

For the proof we reduce from a geometrical problem, the *hypercube problem*, which we show to be NP-hard. Given $d \in \mathbb{N}$, a finite set $P = \{p_1, \ldots, p_k\} \subseteq [0,1]^d$

of vectors ("points") within the d-dimensional unit hypercube, and $\ell \in \mathbb{N}$, the *hypercube problem* asks whether there is a set $Q = \{q_1, \ldots, q_\ell\} \subseteq [0,1]^d$ of at most ℓ points within the hypercube such that $conv(Q) \supseteq P$, where

$$conv(Q) := \{\lambda_1 q_1 + \cdots + \lambda_\ell q_\ell \mid \lambda_1, \ldots, \lambda_\ell \geq 0, \; \lambda_1 + \cdots + \lambda_\ell = 1\}$$

denotes the convex hull of Q. Geometrically, the convex hull of P can be viewed as a convex polytope, nested inside the hypercube, which is another convex polytope. The hypercube problem asks whether a convex polytope with at most ℓ vertices can be nested in between those polytopes. The answer is trivially yes, if $\ell \geq k$ (take $Q = P$) or if $\ell \geq 2^d$ (take $Q = \{0,1\}^d$). We speak of the *restricted hypercube problem* if P contains the origin $(0, \ldots, 0)$. We prove the following:

Proposition 6. *The restricted hypercube problem can in polynomial time be reduced to the PA minimisation problem.*

Proof (sketch). Let $d \in \mathbb{N}$ and $P = \{p_1, \ldots, p_k\} \subseteq [0,1]^d$ and $\ell \in \mathbb{N}$ be an instance of the restricted hypercube problem, where $p_1 = (0, \ldots, 0)$ and $\ell \geq 1$. We construct in polynomial time a PA $\mathcal{A} = (k+1, \Sigma, M, \alpha, \eta)$ such that there is a set $Q = \{q_1, \ldots, q_\ell\} \subseteq [0,1]^d$ with $conv(Q) \supseteq P$ if and only if there is a PA $\mathcal{A}' = (\ell+1, \Sigma, M', \alpha', \eta')$ equivalent to $\mathcal{A}$. Take $\Sigma := \{a_2, \ldots, a_k\} \cup \{b_1, \ldots, b_d\}$. Set $M(a_i)[1, i] := 1$ and $M(b_s)[i, k+1] := p_i[s]$ for all $i \in \{2, \ldots, k\}$ and all $s \in \mathbb{N}_d$, and set all other entries of M to 0. Set $\alpha := e(1)$ and $\eta := e(k+1)^T$. Figure 2 shows an example of this reduction. We prove the correctness of this reduction in [19]. $\qquad \square$
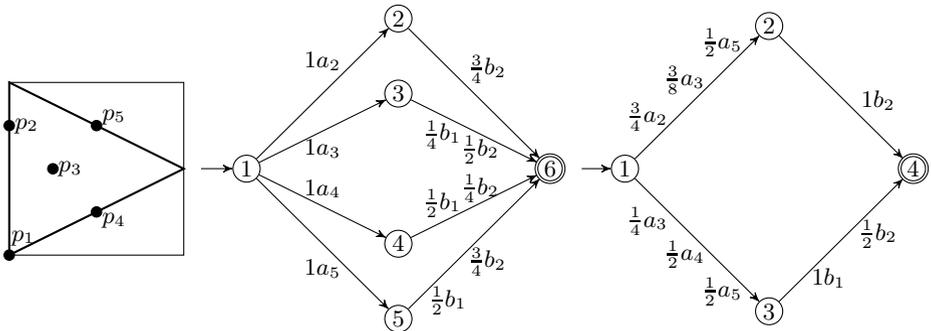

Fig. 2. Reduction from the hypercube problem to the minimisation problem. The left figure shows an instance of the hypercube problem with $d = 2$ and $P = \{p_1, \ldots, p_5\} = \{(0,0), (0, \frac{3}{4}), (\frac{1}{4}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{4}), (\frac{1}{2}, \frac{3}{4})\}$. It also suggests a set $Q = \{(0,0), (0,1), (1, \frac{1}{2})\}$ with $conv(Q) \supseteq P$. The middle figure depicts the PA $\mathcal{A}$ obtained from P. The right figure depicts a minimal equivalent PA $\mathcal{A}'$, corresponding to the set Q suggested in the left figure.

Next we show that the hypercube problem is NP-hard, which together with Proposition 6 implies Theorem 5. A related problem is known[4] to be NP-hard:

Theorem 7 (Theorem 4.2 of [10]). *Given two nested convex polyhedra in three dimensions, the problem of nesting a convex polyhedron with minimum faces between the two polyhedra is NP-hard.*

Note that this NP-hardness result holds even in $d = 3$ dimensions. However, the outer polyhedron is not required to be a cube, and the problem is about minimising the number of faces rather than the number of vertices. Using a completely different technique we show:

Proposition 8. *The hypercube problem is NP-hard. This holds even for the restricted hypercube problem.*

The proof is by a reduction from 3SAT, see [19].

Remark 9. The hypercube problem is in PSPACE, by appealing to decision algorithms for $ExTh(\mathbb{R})$, the existential fragment of the first-order theory of the reals. For every fixed d the hypercube problem is[5] in P, exploiting the fact that $ExTh(\mathbb{R})$ can be decided in polynomial time, if the number of variables is fixed. (For $d = 2$ an efficient algorithm is provided in [1].) It is an open question whether the hypercube problem is in NP. It is also open whether the search for a minimum Q can be restricted to sets of points with rational coordinates (this holds for $d = 2$).

Propositions 6 and 8 together imply Theorem 5.

4.2   Reduction to the Existential Theory of the Reals

In this section we reduce the PA minimisation problem to $ExTh(\mathbb{R})$, the existential fragment of the first-order theory of the reals. A formula of $ExTh(\mathbb{R})$ is of the form $\exists x_1 \ldots \exists x_m R(x_1, \ldots, x_n)$, where $R(x_1, \ldots, x_n)$ is a boolean combination of comparisons of the form $p(x_1, \ldots, x_n) \sim 0$, where $p(x_1, \ldots, x_n)$ is a multivariate polynomial and $\sim \in \{<, >, \leq, \geq, =, \neq\}$. The validity of closed formulas $(m = n)$ is decidable in PSPACE [8,24], and is not known to be PSPACE-hard.

Proposition 10. *Let* $\mathcal{A}_1 = (n_1, \Sigma, M_1, \alpha_1, \eta_1)$ *be a PA. A PA* $\mathcal{A}_2 = (n_2, \Sigma, M_2, \alpha_2, \eta_2)$ *is equivalent to* $\mathcal{A}_1$ *if and only if there exist matrices* $\overrightarrow{M}(a) \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2)}$ *for* $a \in \Sigma$ *and a matrix* $F \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2)}$ *such that* $F[1, \cdot] = (\alpha_1, \alpha_2)$, *and* $F(\eta_1^T, -\eta_2^T)^T = (0, \ldots, 0)^T$, *and*

$$F \begin{pmatrix} M_1(a) & 0 \\ 0 & M_2(a) \end{pmatrix} = \overrightarrow{M}(a)F \qquad \text{for all } a \in \Sigma.$$

The proof is in [19]. The conditions of Proposition 10 on $\mathcal{A}_2$, including that it be a PA, can be phrased in $ExTh(\mathbb{R})$. Thus it follows:

---

[4] The authors thank Joseph O'Rourke for pointing out [10].
[5] This observation is in part due to Radu Grigore.

Theorem 11. *The PA minimisation problem can be reduced in polynomial time to ExTh($\mathbb{R}$). Hence, PA minimisation is in PSPACE.*

Theorem 11 improves on a result in [20] where the minimisation problem was shown to be in EXPTIME. (More precisely, Theorem 4 of [20] states that a minimal PA can be computed in EXPSPACE, but the proof reveals that the decision problem can be solved in EXPTIME.)

5   Conclusions and Open Questions

We have developed a numerically stable and efficient algorithm for minimising WAs, based on linear algebra and Brzozowski-like automata minimisation. We have given bounds on the minimisation error in terms of both the machine epsilon and the error tolerance parameter τ.

We have shown NP-hardness for PA minimisation, and have given a polynomial-time reduction to *ExTh($\mathbb{R}$)*. Our work leaves open the precise complexity of the PA minimisation problem. The authors do not know whether the search for a minimal PA can be restricted to PAs with rational numbers. As stated in the Remark after Proposition 8, the corresponding question is open even for the hypercube problem. If rational numbers indeed suffice, then an NP algorithm might exist that guesses the (rational numbers of the) minimal PA and checks for equivalence with the given PA. Proving PSPACE-hardness would imply PSPACE-hardness of *ExTh($\mathbb{R}$)*, thus solving a longstanding open problem.

For comparison, the corresponding minimisation problems involving WAs (a generalisation of PAs) and DFAs (a special case of PAs) lie in P. More precisely, minimisation of WAs (with rational numbers) is in randomised NC [18], and DFA minimisation is NL-complete [9]. NFA minimisation is PSPACE-complete [16].

References

1. Aggarwal, A., Booth, H., O'Rourke, J., Suri, S., Yap, C.K.: Finding minimal convex nested polygons. Information and Computation 83(1), 98–110 (1989)
2. Arnoldi, W.E.: The principle of minimized iteration in the solution of the matrix eigenvalue problem. Quarterly of Applied Mathematics 9, 17–29 (1951)
3. Beimel, A., Bergadano, F., Bshouty, N.H., Kushilevitz, E., Varricchio, S.: Learning functions represented as multiplicity automata. Journal of the ACM 47(3), 506–530 (2000)
4. Berstel, J., Reutenauer, C.: Rational Series and Their Languages. Springer (1988)
5. Bonchi, F., Bonsangue, M.M., Hansen, H.H., Panangaden, P., Rutten, J.J.M.M., Silva, A.: Algebra-coalgebra duality in Brzozowski's minimization algorithm. ACM Transactions on Computational Logic (to appear)

6. Brzozowski, J.A.: Canonical regular expressions and minimal state graphs for definite events. In: Symposium on Mathematical Theory of Automata. MRI Symposia Series, vol. 12, pp. 529–561. Polytechnic Press, Polytechnic Institute of Brooklyn (1962)

7. Bukharaev, R.G.: Probabilistic automata. Journal of Soviet Mathematics 13(3), 359–386 (1980)

8. Canny, J.: Some algebraic and geometric computations in PSPACE. In: Proceedings of STOC 1988, pp. 460–467 (1988)

9. Cho, S., Huynh, D.T.: The parallel complexity of finite-state automata problems. Information and Computation 97(1), 1–22 (1992)

10. Das, G., Goodrich, M.T.: On the complexity of approximating and illuminating three-dimensional convex polyhedra. In: Sack, J.-R., Akl, S.G., Dehne, F., Santoro, N. (eds.) WADS 1995. LNCS, vol. 955, pp. 74–85. Springer, Heidelberg (1995)

11. Das, G., Joseph, D.: Minimum vertex hulls for polyhedral domains. Theoretical Computer Science 103(1), 107–135 (1992)

12. Golub, G.H., van Loan, C.F.: Matrix Computations. John Hopkins University Press (1989)

13. Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 2nd edn. SIAM (2002)

14. Householder, A.S.: Unitary triangularization of a nonsymmetric matrix. Journal of the ACM 5(4), 339–342 (1958)

15. Culik II, K., Kari, J.: Image compression using weighted finite automata. Computers & Graphics 17(3), 305–313 (1993)

16. Jiang, T., Ravikumar, B.: Minimal NFA problems are hard. SIAM Journal on Computing 22(6), 1117–1141 (1993)

17. Kiefer, S., Murawski, A.S., Ouaknine, J., Wachter, B., Worrell, J.: Language equivalence for probabilistic automata. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 526–540. Springer, Heidelberg (2011)

18. Kiefer, S., Murawski, A.S., Ouaknine, J., Wachter, B., Worrell, J.: On the complexity of equivalence and minimisation for Q-weighted automata. Logical Methods in Computer Science 9(1:8), 1–22 (2013)

19. Kiefer, S., Wachter, B.: Stability and complexity of minimising probabilistic automata. Technical report, arxiv.org (2014), http://arxiv.org/abs/1404.6673

20. Mateus, P., Qiu, D., Li, L.: On the complexity of minimizing probabilistic and quantum automata. Information and Computation 218, 36–53 (2012)

21. Mitchell, J.S.B., Suri, S.: Separation and approximation of polyhedral objects. In: Proceedings of SODA, pp. 296–306 (1992)

22. Paz, A.: Introduction to probabilistic automata. Academic Press (1971)

23. Rabin, M.O.: Probabilistic automata. Information and Control 6(3), 230–245 (1963)

24. Renegar, J.: On the computational complexity and geometry of the first-order theory of the reals. Parts I–III. Journal of Symbolic Computation 13(3), 255–352 (1992)

25. Schützenberger, M.-P.: On the definition of a family of automata. Information and Control 4, 245–270 (1961)

26. Silio, C.B.: An efficient simplex coverability algorithm in E^2 with application to stochastic sequential machines. IEEE Transactions on Computers C-28(2), 109–120 (1979)