

Efficient Tests for Equivalence of Hidden Markov Processes and Quantum Random Walks

Ulrich Faigle¹ and Alexander Schönhuth², Member IEEE

¹ Mathematical Institute
Center for Applied Computer Science
University of Cologne
Weyertal 80
50931 Köln, Germany

² Centrum Wiskunde & Informatica
Science Park 123
1098 XG Amsterdam
as@cw.i.nl

Abstract. While two hidden Markov process (HMP) resp. quantum random walk (QRW) parametrizations can differ from one another, the stochastic processes arising from them can be equivalent. Here a polynomial-time algorithm is presented which can determine equivalence of two HMP parametrizations $\mathcal{M}_1, \mathcal{M}_2$ resp. two QRW parametrizations $\mathcal{Q}_1, \mathcal{Q}_2$ in time $O(|\Sigma| \max(N_1, N_2)^4)$, where N_1, N_2 are the number of hidden states in $\mathcal{M}_1, \mathcal{M}_2$ resp. the dimension of the state spaces associated with $\mathcal{Q}_1, \mathcal{Q}_2$, and Σ is the set of output symbols. Previously available algorithms for testing equivalence of HMPs were exponential in the number of hidden states. In case of QRWs, algorithms for testing equivalence had not yet been presented. The core subroutines of this algorithm can also be used to efficiently test hidden Markov processes and quantum random walks for ergodicity.

Keywords. Dimension, Discrete Random Sources, Hidden Markov Processes, Identifiability, Linearly Dependent Processes, Quantum Random Walks,

1 Introduction

Let a parameterized class of stochastic processes be described by a mapping

$$\Phi : \mathcal{P} \rightarrow \mathcal{S} \quad (1)$$

where \mathcal{P} is the set of parameterizations and \mathcal{S} is the corresponding set of stochastic processes. A stochastic process $\Phi(P)$ as induced by the parameterization P is said to be *identifiable* iff

$$\Phi^{-1}(\Phi(P)) = \{P\} \quad (2)$$

that is, iff the parameterization giving rise to it is uniquely determined. The entire class of stochastic processes $\Phi(\mathcal{P})$ is said to be *identifiable* iff $\Phi : \mathcal{P} \rightarrow \Phi(\mathcal{P})$ is one-to-one. The *equivalence problem* (EP) emerges when Φ is many-to-one and is to decide

whether two parameterizations P_1, P_2 are equivalent, that is $\Phi(P_1) = \Phi(P_2)$. Understanding its solutions can significantly foster understanding of the classes of stochastic processes under consideration as it usually yields insights about the class' complexity and its number of free parameters. Therefore, apart from its theoretical relevance, it is an important issue in the practice of system identification (e.g. [18]).

Hidden Markov processes (HMPs) are a class of processes which have gained widespread attention. In practical applications, for example, they have established gold standards in speech recognition and certain areas of computational biology. See e.g. [19,6,7] for comprehensive related literature. In an intuitive description, a hidden Markov process is governed by a Markov process which, however, cannot be observed. Observed symbols are emitted according to another set of distributions which govern the hidden, non-observed states. Since observed processes can coincide although the non-observed processes on the hidden states can differ from one another, hidden Markov processes are non-identifiable.

For hidden Markov processes, the EP was first discussed in 1957 [3] (see also [11] for a subsequent contribution). It was formulated for finite functions of Markov chains (FFMCs), an alternative way of parametrizing hidden Markov processes where, as sets of parametrizations, the parametrizations discussed here, also referred to as *hidden Markov models (HMMs)* in the following, models trivially contain FFMCs. The EP for hidden Markov processes was fully solved in 1992 [13]. The corresponding algorithm is exponential in the number of hidden states and therefore impractical for larger models. See [13] also for more related work.

Quantum random walks (QRWs) have been introduced to quantum information theory as an analog of classical Markov sources [1]. For example, they allow emulation of Markov Chain Monte Carlo approaches on quantum computers. A collection of results has pointed out that they would be superior to their classical counterparts with respect to a variety of aspects (see e.g. [15,2,16]). However, although their mechanisms can be described in terms of elementary linear algebraic definitions, their properties are much less understood. The key element of a quantum random walk parametrization is a graph whose vertices are the observed symbols. Quantum probability distributions on the vertices are transformed by linear operations which describe the quantum mechanical concepts of evolution and measurement. It is easy to see that quantum random walks are non-identifiable. For example, any of the (infinitely many different) parametrizations with a graph of only one vertex yields the same, trivial process. The equivalence problem for quantum random walks has not been discussed before.

Beyond the work cited in [13], there is a polynomial-time solution to test equivalence of probabilistic automata [21] where HMMs can be viewed as probabilistic automata with no final probabilities [5]. The crucial difference, however, is that probabilistic automata do not give rise to stochastic processes (distributions over infinite-length sequences), but to probability distributions over the set of strings of finite length. The algorithm presented in [21] decisively depends on this and therefore does neither apply for hidden Markov processes nor for quantum random walks. Conversely, by adding a stop symbol to the set of observed symbols, any probability distribution over the set of strings of finite length can be viewed as a probability distribution over the set of infinite-length symbol sequences. This way, it can be seen that our solution also applies

for probabilistic automata and therefore is more general than [21]’s solution.

Overall, the purpose of this work is to present a simple, polynomial-time algorithm that solves the EP for both hidden Markov processes and quantum random walks:

Theorem 1. *Let Σ be a finite set of symbols and*

$$\mathcal{M}_X, \mathcal{M}_Y \quad \text{resp.} \quad \mathcal{Q}_X, \mathcal{Q}_Y \quad (3)$$

be two hidden Markov process resp. quantum random walk parametrizations giving rise to the processes $(X_t), (Y_t)$ emitting symbols from Σ . Let

$$n_X, n_Y \quad (4)$$

be the cardinalities of the set of hidden states in the hidden Markov models resp. the dimensions of the state spaces associated with the quantum random walks. Equivalence of $(X_t), (Y_t)$ can be determined in

$$O(|\Sigma| \max\{n_X, n_Y\}^4) \quad (5)$$

arithmetic operations.

Remark 1. Note that a polynomial-time solution for the identifiability problem for HMPs does not provide a polynomial-time solution for the graph isomorphism problem. There are both non-equivalent HMPs which act on sets of hidden states which are isomorphic as graphs (e.g. two HMPs both acting on only one hidden state which, however, have different emission probability distributions) and equivalent HMPs where underlying graphs are non-isomorphic (e.g. two HMPs, one acting on two hidden states, but emitting the symbol a with probability 1 from both states and the other one acting on only one hidden state, also emitting the symbol a with probability 1, both result in the stochastic process which generates $aaaa\dots$ with probability 1).

Remark 2. In [20] it was described how to test HMPs for ergodicity. Plugging the algorithm for computation of a basis (see subsection 5.1) into the generic ergodicity tests provided in [20] renders these tests efficient.

1.1 Organization of Sections

The core ideas of this work are tightly interconnected with the theory of *finitary processes*. Therefore, we start by concisely revisiting their theory in section 2. We then introduce hidden Markov models and quantum random walk parametrizations and the mechanisms which give rise to the associated processes in section 3. In section 4 we outline how to most efficiently compute probabilities for both hidden Markov processes and quantum random walks. The algorithm and theorems behind our efficient equivalence tests are then presented in section 5. We finally outline some complementary applications of our algorithms and make some conclusive remarks.

2 Finitary Random Processes

Throughout this paper, we consider discrete random processes (X_t) that take values in the (fixed) finite alphabet Σ . We assume that the process emits the *empty word* \square at time $t = 0$. We denote the *probability function* p of (X_t) by

$$p(a_1 \dots a_t) := \Pr\{X_1 = a_1, \dots, X_t = a_t\} \quad (a_1 \dots a_t \in \Sigma^t). \quad (6)$$

As usual, we set

$$\Sigma^* := \bigcup_{t \geq 0} \Sigma^t \quad (\text{with } \Sigma^0 = \{\square\}) \quad (7)$$

and note that Σ^* is a semigroup under the concatenation $wv \in \Sigma^{s+t}$ for $w \in \Sigma^s$ and $v \in \Sigma^t$. $|v| = \ell$ is the *length* of a word $v \in \Sigma^\ell$.

For any $v, w \in \Sigma^*$, we define functions $p_v, p^w : \Sigma^* \rightarrow \mathbb{R}$ via

$$p_v(w) := p(wv) =: p^w(v) \quad (8)$$

and view \mathbb{R}^{Σ^*} as a vector space. $p(v = v_1 \dots v_t | w = w_1 \dots w_s)$ generally denotes the *conditional probability*

$$\begin{aligned} p(v|w) &:= \Pr(X_{s+1} = v_1, \dots, X_{t+s} = v_t \mid X_1 = w_1, \dots, X_s = w_s) \\ &= \begin{cases} p(v) & \text{if } w = \square \\ 0 & \text{if } p(w) = 0 \\ p(wv)/p(w) & \text{otherwise.} \end{cases} \end{aligned} \quad (9)$$

Furthermore, the subspace

$$\mathcal{R}(p) := \text{span}\{p_v \mid v \in \Sigma^*\} \quad \text{resp.} \quad \mathcal{C}(p) := \text{span}\{p^w \mid w \in \Sigma^*\} \quad (10)$$

is the *row space* resp. *column space* associated with the (probability function p of) the random process (X_t) .

It is easy to see that $\mathcal{R}(p)$ and $\mathcal{C}(p)$ have the same vector space dimension. So we define the *dimension* of (X_t) (or its probability function p) as the parameter

$$\dim(X_t) = \dim(p) := \dim \mathcal{R}(p) = \dim \mathcal{C}(p) \in \mathbb{Z}_+ \cup \{\infty\}. \quad (11)$$

For any $I, J \subseteq \Sigma^*$, we define the matrix

$$P_{IJ} := [p(wv)]_{v \in I, w \in J} \in \mathbb{R}^{I \times J}. \quad (12)$$

P_{IJ} is called *generating* if $\text{rk}(P_{IJ}) = \text{rk}(P_{\Sigma^*, \Sigma^*}) (= \dim(p))$ and *basic* if it is generating and minimal among the generating P_{IJ} , that is $|I| = |J| = \dim(p)$ in case of $\dim(p) < \infty$. In that sense, we call (in slight abuse of language) I resp. J a *row* resp. *column generator/basis* and the pair (IJ) a *generator/basis* for p .

We call a process (X_t) *finitary* if it admits a (finite) basis. So the finitary processes are exactly the ones with finite dimension.

Remark 3. The dimension of a random process is known as its *minimum degree of freedom*. The term *finitary* was introduced in [12]. Finitary processes are also called *linearly dependent* [14].

Theorem 2. Let (X_t) and (Y_t) be discrete, finitary random processes (over Σ) with probability functions p and q . Let furthermore (IJ) be a basis for (X_t) . Then the following statements are equivalent:

- (a) $p = q$.
- (b) (I, J) is a basis for (Y_t) and the equalities

$$p(v) = q(v), \quad p(wv) = q(wv) \quad \text{and} \quad p(wav) = q(wav) \quad (13)$$

hold for all choices of $v \in I, w \in J$ and $a \in \Sigma$.

Proof. Given a basic matrix P_{IJ} together with the probabilities $p(v), p(wav)$ for all $v \in I, w \in J, a \in \Sigma$, one can reconstruct p via a "minimal representation" (see, e.g., [13,14,20] for details). \diamond

3 Parametrizations and The Equivalence Problem

3.1 Hidden Markov Processes

A *hidden Markov process (HMP)* is parametrized by a tuple $\mathcal{M} = (S, E, \pi, M)$ where

1. $S = \{s_1, \dots, s_n\}$ is a finite set of "hidden" states
2. $E = [e_{sa}] \in \mathbb{R}^{S \times \Sigma}$ is a non-negative *emission probability matrix* with unit row sums $\sum_{a \in \Sigma} E_{sa} = 1$, (i.e. the row vectors of E are probability distributions on Σ)
3. π is an *initial probability distribution* on S and
4. $M = [m_{ij}] \in \mathbb{R}^{S \times S}$ is a non-negative *transition probability matrix* with unit row sums $\sum_{j=1}^n m_{ij} = 1$ (i.e. the row vectors of M are probability distributions on S)

The associated process (X_t) initially moves to a state $s \in S$ with probability π_s and emits the symbol $X_1 = a$ with probability E_{sa} . Then it moves from s to a state s' with probability $m_{ss'}$ and emits the symbol $X_2 = a'$ with probability $E_{s'a'}$ and so on. In the following, we also refer to a parametrization $\mathcal{M} = (S, E, \pi, M)$ as a *hidden Markov model (HMM)*.

3.2 Quantum Random Walks

A *quantum random walk (QRW)* is parametrized by a tuple $\mathcal{Q} = (G, U, \psi_0)$ where

1. $G = (\Sigma, E)$ is a directed, K -regular graph over the alphabet Σ
2. $U : \mathbb{C}^k \rightarrow \mathbb{C}^k$ is a unitary *evolution* operator where $k := |E| = K \cdot |\Sigma|$ and
3. $\psi_0 \in \mathbb{C}^k$ is a wave function, that is $\|\psi_0\| = 1$ ($\|\cdot\|$ the Euclidean norm).

Edges are labeled by tuples (a, x) , $a \in \Sigma$, $x \in X$ where X is a finite set with $|X| = k$. Correspondingly, \mathbb{C}^k is considered to be spanned by the orthonormal basis

$$\langle \mathbf{e}_{(a,x)} \mid (a,x) \in E \rangle.$$

According to [1] some more specific conditions must hold which do not affect our considerations here.

The *quantum random walk* (X_t) arising from a parametrization $\mathcal{Q} = (G, U, \psi_0)$ proceeds by first applying the unitary operator U to ψ_0 and subsequently, with probability $\sum_{x \in X} |(U\psi_0)_{(a,x)}|^2$, “collapsing” (i.e. projecting and renormalizing, which models a quantum mechanical measurement) $U\psi_0$ to the subspace spanned by the vectors $\mathbf{e}_{(a,x), x \in X}$ to generate the first symbol $X_1 = a$. Collapsing $U\psi_0$ results in a new wave function ψ_1 . Applying U to ψ_1 and collapsing it, with probability $\sum_{x \in X} |(U\psi_1)_{(a',x)}|^2$, to the subspace spanned by $\mathbf{e}_{(a',x), x \in X}$ generates the next symbol $X_2 = a'$. Iterative application of U and subsequent collapsing generates further symbols.

3.3 The Equivalence Problem

The *equivalence problem* can be framed as follows:

Equivalence Problem (IP)

Given two hidden Markov models $\mathcal{M}_X, \mathcal{M}_Y$ or two quantum random walk parametrizations $\mathcal{Q}_X, \mathcal{Q}_Y$, decide whether the associated processes (X_t) and (Y_t) are equivalent.

The equivalence problem can, of course, be solved in principle, in the spirit of Theorem 2. In order to efficiently solve it in practice, it suffices to be able to efficiently compute the following quantities:

- (1) A basis (I, J) for the finitary processes $(X_t), (Y_t)$ from their parametrizations $\mathcal{M}_X, \mathcal{M}_Y$.
- (2) The corresponding probabilities $p(v), p(wv), p(wav)$ for all choices of $v \in I, w \in J, a \in \Sigma$.

4 Computing Probabilities

We would like to point out that in the following we assume that all inputs consist of rational numbers and that each arithmetic operation can be done in constant time. This agrees with the usual conventions when treating related probabilistic concepts in terms of algorithmic complexity [19,21,5].

4.1 Hidden Markov Processes

Let now (X_t) be a hidden Markov process with parametrization $\mathcal{M} = (S, E, \pi, M)$. Observe first that the *transition matrix* M decomposes as $M = \sum_{a \in \Sigma} T_a$ into matrices T_a with coefficients

$$(T_a)_{ij} := e_{s_i a} \cdot m_{ij} \quad (14)$$

which reflect the probabilities to emit symbol a from state s_i and subsequently to move on to state s_j . Standard technical computations (e.g. [8]) reveal that that for any word $a_1 \dots a_t \in \Sigma^t$:

$$\begin{aligned} p(a_1 a_2 \dots a_t) &= p(a_1 a_2 \dots a_{t-1}) p(a_t | a_1 a_2 \dots a_{t-1}) \\ &= \dots \\ &= \pi^T T_{a_1} \dots T_{a_{t-1}} T_{a_t} \mathbf{1}, \end{aligned} \quad (15)$$

where $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^S$ is the vector of all ones.

For further reference, we use the notations

$$T_v := T_{v_1} T_{v_2} \dots T_{v_{t-1}} T_{v_t} \in \mathbb{R}^{n \times n} \quad (16)$$

for any $v = v_1 \dots v_t \in \Sigma^*$ as well as

$$\vec{\mathbf{p}}(v) := \pi^T T_v \in \mathbb{R}^{1 \times n} \quad \text{and} \quad \overleftarrow{\mathbf{p}}(v) := T_v \mathbf{1} \in \mathbb{R}^{n \times 1}. \quad (17)$$

Remark 4. Note that computation of vectors $\vec{\mathbf{p}}(v)$ and $\overleftarrow{\mathbf{p}}(v)$ is an alternative way to describe the well-known Forward and Backward algorithm (e.g. [7]) since the entries of these two vectors can be identified with the Forward and Backward variables

$$\Pr(S_{s+1} = s_i \mid X_1 = a_1, \dots, X_s = a_s) \quad (18)$$

$$\text{and} \quad (19)$$

$$\Pr(S_{s+1} = s_i \mid X_{s+1} = a_{s+1}, \dots, X_{s+t} = a_{s+t}) \quad (20)$$

where (S_t) is the (non-observable) Markov process over the hidden states $S = (s_1, \dots, s_n)$.

4.2 Quantum Random Walks

The following considerations can be straightforwardly derived from standard quantum mechanical arguments, see [4] for a reference.

The State Space \mathcal{S}^n We write Q^* for the adjoint of an arbitrarily sized matrix $Q \in \mathbb{C}^{m \times n}$, (that is $Q_{ji}^* = a - ib$ if $Q_{ji} = a + ib$ where usage of i as both a running index and a complex number should not lead to confusion). Let

$$n := k^2 = |E|^2. \quad (21)$$

We will consider the set of self-adjoint matrices

$$\mathcal{S}^n := \{Q \in \mathbb{C}^{k^2} \mid Q = Q^*\} \quad (22)$$

in the following, which is usually referred to as *state space* in quantum mechanics. As usual, \mathcal{S}^n can be viewed as an $n = k^2$ -dimensional real-valued vector space. To illustrate this let

$$\mathbf{e}_m := (0, \dots, 0, \underset{m}{1}, 0, \dots, 0)^T \in \mathbb{C}^K, m = 1, \dots, k \quad \text{and} \quad (23)$$

$$\mathbf{f}_m := (0, \dots, 0, \underset{m}{i}, 0, \dots, 0)^T \in \mathbb{C}^K, m = 1, \dots, k. \quad (24)$$

The self-adjoint matrices

$$\mathbf{E}_{m_1 m_2} := (\mathbf{e}_{m_1} \mathbf{e}_{m_2}^* + \mathbf{e}_{m_2} \mathbf{e}_{m_1}^*) \quad \text{and} \quad \mathbf{F}_{m_1 m_2} := (\mathbf{f}_{m_1} \mathbf{f}_{m_2}^* + \mathbf{f}_{m_2} \mathbf{f}_{m_1}^*) \quad (25)$$

for all choices of $1 \leq m_1, m_2 \leq k$ and $m_1 \neq m_2$ for $\mathbf{F}_{m_1 m_2}$ (since entries on the diagonal of self-adjoint matrices are real-valued) then form a canonical basis of \mathcal{S}^n (note that $\mathbf{E}_{m_1 m_2} = \mathbf{E}_{m_2 m_1}$, $\mathbf{F}_{m_1 m_2} = \mathbf{F}_{m_2 m_1}$).

Linear Operations on \mathcal{S}^n For a quantum random walk parametrization $\mathcal{Q} = (G = (\Sigma, E), U, \psi_0)$ we introduce the projection operators ($k := |E|$)

$$P_a : \mathbb{C}^k \longrightarrow \mathbb{C}^k, \psi \mapsto \sum_{(a,x), x \in X} \psi_{(a,x)} \mathbf{e}_{(a,x)} \quad (26)$$

for all $a \in \Sigma$ which reflects projection of ψ onto the subspace spanned by the $\mathbf{e}_{(a,x)}$, $x \in X$. We find that

$$T_a : \mathcal{S}^n \longrightarrow \mathcal{S}^n, Q \mapsto (P_a U) Q (P_a U)^* \quad (27)$$

is an \mathbb{R} -linear operator acting on the state space \mathcal{S}^n . In analogy to the theory of hidden Markov models, where here the order on the letters has been reversed, we further define

$$T_v := T_{v_t} T_{v_{t-1}} \dots T_{v_2} T_{v_1} \in \mathbb{R}^{n \times n} \quad (28)$$

for any $v = v_1 \dots v_t \in \Sigma^*$.

Let now $Q_\psi := \psi \psi^* \in \mathbb{C}^{k \times k}$ be the self-adjoint matrix being associated with a wave function $\psi \in \mathbb{C}^k$. We recall that, by definition of the quantum random walk p with parametrization \mathcal{Q} , probabilities $p(v = v_1 \dots v_t)$ are computed as

$$p(v = v_1 \dots v_t) = \|(P_{v_t} U)(P_{v_{t-1}}) \dots (P_{v_1} U) \psi_0\|^2 \quad (29)$$

which can be rephrased as ($Q_{\psi_0} := \psi_0 \psi_0^*$ and tr is the linear trace functional, that is the sum of the diagonal entries)

$$p(v = v_1 \dots v_t) = \text{tr } T_{v_t} \dots T_{v_1} Q_{\psi_0} \quad (30)$$

which yields that probabilities $p(v)$ can be computed by iterative application of multiplying $n \times n$ -matrices with n -dimensional vectors where we recall that Q_{ψ_0} can be taken as an element of the n -dimensional vector space \mathcal{S}^n . Note that T_v acts on Q_{ψ_0} in the sense of \mathcal{S}^n whereas the trace functional treats $T_v Q_{\psi_0}$ as a matrix.

Forward and Backward Algorithm Note that application of the trace functional can be rephrased as

$$\text{tr } Q = E \cdot Q \in \mathbb{R} \quad \text{where} \quad E := \sum_{i=1}^k \mathbf{e}_i \mathbf{e}_i^* \quad (31)$$

and, on the right hand side, both E and Q are taken as elements of \mathcal{S}^n , i.e. as n -dimensional vectors. Using this, we define

$$\vec{\mathbf{p}}(v) := T_v Q_{\psi_0} \in \mathcal{S}^n \subset \mathbb{C}^{n^2} \quad \text{and} \quad \overleftarrow{\mathbf{p}}(v) := E T_v \in \mathbb{C}^{n^2}. \quad (32)$$

Computation of $\vec{\mathbf{p}}(v)$ and $\overleftarrow{\mathbf{p}}(v)$ can be taken as performing a quantum random walk version of the Forward and the Backward algorithm. Correspondingly, entries of $\vec{\mathbf{p}}(v)$ and $\overleftarrow{\mathbf{p}}(v)$ reflect Forward and Backward variables.

4.3 Runtimes

Since the multiplication of an $(n \times n)$ -matrix with a vector can be done in $O(n^2)$ arithmetic operations, the previous considerations let us conclude:

Lemma 1. *Given \mathcal{M} or \mathcal{Q} let n be the number of hidden states $|S|$ resp. the dimension of the state space \mathcal{S}^n associated with \mathcal{Q} and p be the probability function of \mathcal{M} or \mathcal{Q} .*

1. *For any $v \in \Sigma^*$*

$$\vec{\mathbf{p}}(v), \overleftarrow{\mathbf{p}}(v) \quad \text{and} \quad p(v) \quad (33)$$

can be computed in $O(|v|n^2)$ arithmetic operations.

2. *Upon computation of $\vec{\mathbf{p}}(w)$ computation of all*

$$p(wa) \quad \text{and} \quad \vec{\mathbf{p}}(wa) \quad (34)$$

requires $O(|\Sigma|n^2)$ arithmetic operations.

3. *Upon computation of $\overleftarrow{\mathbf{p}}(v)$ computation of all*

$$p(av) \quad \text{and} \quad \overleftarrow{\mathbf{p}}(av) \quad (35)$$

requires $O(|\Sigma|n^2)$ arithmetic operations.

4. *Upon computation of $\vec{\mathbf{p}}(w)$ and $\overleftarrow{\mathbf{p}}(v)$ computation of all*

$$p(wav) \quad (36)$$

requires $O(|\Sigma|n^2)$ arithmetic operations. \diamond

For hidden Markov models \mathcal{M} this actually reflects well-known results on computation of Forward/Backward variables.

5 Equivalence Tests

In this section, we describe how to efficiently test two hidden Markov processes or quantum random walks (X_t) and (Y_t) for equivalence. We recall that a generic strategy has been established by theorem 2. Our solution proceeds according to this strategy.

5.1 Computation of a Basis

We will now show how to compute a basis (IJ) in runtime $O(|\Sigma|n^4)$ for a hidden Markov process resp. a quantum random walk p . Therefore, assume for now that $g_1, \dots, g_n : \Sigma^* \rightarrow \mathbb{R}$ are probability functions the probabilities of which can be computed in the style of hidden Markov processes resp. quantum random walks and which generate the column space of p , i.e.,

$$\mathcal{C}(p) \subset \text{span}\{g_1, \dots, g_n\}. \quad (37)$$

Given g_1, \dots, g_n , computation of a basis (IJ) proceeds in three steps the first two of which are analogous and the third of which is a simple procedure.

1. Compute a row generator I .
2. Compute a column basis J .
3. Reduce I to a row basis.

While steps 1 and 2 both require runtime $O(|\Sigma|n^4)$, step 3 requires $O(n^4)$ which overall evaluates as $O(|\Sigma|n^4)$ runtime required for computation of a basis.

We discuss the steps in the following paragraphs. In a subsequent subsection, we show how to obtain suitable g_1, \dots, g_n for both hidden Markov models and quantum random walks.

Step 1: Computation of a row generator I Consider the following algorithm.

Algorithm 1

- 1: Define $\mathbf{g}(v) = (g_1(v), \dots, g_n(v)) \in \mathbb{R}^n$.
- 2: $I \leftarrow \{\square\}$, $B_{row} \leftarrow \{\mathbf{g}(\square)\}$, $C_{row} \leftarrow \Sigma$.
- 3: **while** $C_{row} \neq \emptyset$ **do**
- 4: Choose $v \in C_{row}$.
- 5: **if** $\mathbf{g}(v)$ is linearly independent of B_{row} **then**
- 6: $I \leftarrow I \cup \{v\}$, $B_{row} \leftarrow B_{row} \cup \{\mathbf{g}(v)\}$
- 6: $C_{row} \leftarrow C_{row} \cup \{av \mid a \in \Sigma\}$
- 7: **end if**
- 8: **end while**
- 9: **output** I .

Proposition 1. *Let $I \subseteq \Sigma^*$ be the output of Algorithm 1. Then one has*

$$\mathcal{R}(p) = \text{span}\{p_v \mid v \in I\} \quad \text{and} \quad \dim(X_t) \leq |I| \quad (38)$$

where

$$\mathcal{C}(p) = \text{span}\{g_1, \dots, g_n\} \quad \Rightarrow \quad \dim(X_t) = |I|. \quad (39)$$

Furthermore,

- (i) *The algorithm terminates after at most $|\Sigma| \cdot n$ iterations.*

(ii) Each iteration requires $O(n^3)$ arithmetic operations where at most n iterations need additional $O(|\Sigma|n^3)$ operations.

Proof. Ad (i): Because the n -dimensional vectors in B_{row} are independent $|B_{row}| \leq n$ and $|I| \leq n$ follow immediately. Since at most Σ words are added to C_{row} upon discovery of an n -dimensional vector which is linearly independent of those in B_{row} , we have $|C_{row}| \leq |\Sigma| \cdot n$ and hence at most $|\Sigma| \cdot n$ iterations.

Ad (ii): In each iteration, we perform a test for linear independency of at most n vectors of dimension n which requires at most $O(n^3)$ arithmetic operations [10]. In the at most n cases where $\mathbf{g}(v)$ is linearly independent of B_{row} , we proceed by computing

$$(g_1(av), \dots, g_n(av)) \quad \text{and} \quad (\overleftarrow{\mathbf{g}}_1(av), \dots, \overleftarrow{\mathbf{g}}_n(av)) \quad (40)$$

for all $a \in \Sigma$ where $(\overleftarrow{\mathbf{g}}_1(v), \dots, \overleftarrow{\mathbf{g}}_n(v))$ are available from an iteration before (note that $g_i(\square) = 1, \overleftarrow{\mathbf{g}}_i(\square) = (1, \dots, 1)$ in the first iteration). Due to lemma 1, (35), this requires $O(|\Sigma| \cdot n^3)$ operations.

To prove (38), let $w_0 \in \Sigma^*$ be arbitrary and suppose

$$p_{w_0} \notin \text{span}\{p_v \mid v \in I\}. \quad (41)$$

Since $\mathcal{C}(p) \subset \text{span}\{g_1, \dots, g_n\}$, plugging $w = \square$ into lemma 2 below implies

$$\mathbf{g}(w_0) \notin \text{span}\{\mathbf{g}(v) \mid v \in I\}. \quad (42)$$

We will derive a contradiction. Indeed, the algorithm can only miss w_0 if w_0 had never been collected into C_{row} in step 6. This happens only in case that there is a $v_0 \in \Sigma^*$ such that

$$w_0 = wv_0 \quad (43)$$

holds for some $w \in \Sigma^*$ and $\mathbf{g}(v_0)$ had been found to be linearly dependent of $[\mathbf{g}(v)]_{v \in I}$. Lemma 2 below then states that in such a case $p_{w_0} \in \text{span}\{p_{wv} \mid v \in I\}$ holds and it remains to show that for each $w \in \Sigma^*$ and $v \in I$

$$p_{wv} \in \text{span}\{p_v \mid v \in I\}. \quad (44)$$

This follows by induction on the length $|w|$ of w from the following arguments. For each $w \in \Sigma^*$ we define a linear operator σ_w on $\mathcal{R}(p)$ through

$$\sigma_w p_v = p_{wv}. \quad (45)$$

By design of the update rule for C_{row} in step 6 of algorithm 1 we immediately see that

$$\mathbf{g}(av) \in \text{span}\{\mathbf{g}(v) \mid v \in I\} \quad (46)$$

for all $a \in \Sigma$, hence by plugging $v_0 = av$ and $w = \square$ into lemma 2, we obtain $p_{av} \in \text{span}\{p_v \mid v \in I\}$ that is

$$\sigma_a(\text{span}\{p_v \mid v \in I\}) \subset \text{span}\{p_v \mid v \in I\} \quad (47)$$

for all $a \in \Sigma$. Inductively, by observing that $\sigma_{w=w_1 \dots w_t} = \sigma_{w_1} \circ \dots \circ \sigma_{w_t}$,

$$\sigma_w(\text{span}\{p_v \mid v \in I\}) \subseteq \text{span}\{p_v \mid v \in I\} \quad (48)$$

and thereby (44).

To see (39) let $\dim(X_t) < |I|$. Since $|I| = |B_{\text{row}}|$ we obtain that

$$\dim \mathcal{C}(p) = \dim(X_t) < |B_{\text{row}}| \leq \dim \text{span}\{g_1, \dots, g_n\} \quad (49)$$

hence $\mathcal{C}(p) \subsetneq \text{span}\{g_1, \dots, g_n\}$. \diamond

Lemma 2. *Let $g_1, \dots, g_n : \Sigma^* \rightarrow \mathbb{R}$ be such such that $\mathcal{C}(p) \subseteq \text{span}\{g_1, \dots, g_n\}$ and let $v_0, v_1, \dots, v_m \in \Sigma^*$ be such that*

$$(g_1(v_0), \dots, g_n(v_0)) \in \text{span}\{(g_1(v_j), \dots, g_n(v_j)) \mid j = 1, \dots, m\} \subseteq \mathbb{R}^n. \quad (50)$$

Then one has for every $w \in \Sigma^$:*

$$p_{wv_0} \in \text{span}\{p_{wv_j} \mid j = 1, \dots, m\} \subseteq \mathbb{R}^n. \quad (51)$$

The analogous statement holds for the row space $\mathcal{R}(p)$.

Proof. By our hypothesis, there are scalars $\beta_1, \dots, \beta_m \in \mathbb{R}$ such that

$$(g_1(v_0), \dots, g_n(v_0)) = \sum_{j=1}^m \beta_j (g_1(v_j), \dots, g_n(v_j)). \quad (52)$$

Let $u \in \Sigma^*$ be arbitrary. Again by our hypothesis, there are scalars $\alpha_i, i = 1, \dots, n \in \mathbb{R}$ such that

$$p_u = \sum_{i=1}^n \alpha_i g_i. \quad (53)$$

We now compute

$$\begin{aligned} p_u(v_0) &\stackrel{(53)}{=} \sum_{i=1}^n \alpha_i g_i(v_0) \\ &\stackrel{(52)}{=} \sum_{i=1}^n \alpha_i \sum_{j=1}^m \beta_j g_i(v_j) = \sum_{j=1}^m \beta_j \sum_{i=1}^n \alpha_i g_i(v_j) \\ &\stackrel{(53)}{=} \sum_{j=1}^m \beta_j g_u(v_j) = \sum_{j=1}^m \beta_j p(uv_j) = \sum_{j=1}^m \beta_j p_{v_j}(u). \end{aligned}$$

Since the β_j had been determined independently of u , we thus conclude

$$p_{v_0} = \sum_{j=1}^m \beta_j p_{v_j}. \quad (54)$$

Let σ_w be the linear operator on $\mathcal{R}(p)$ with the property

$$\sigma_w p_v = p_{wv}. \quad (55)$$

Application of σ_w to (54) then shows

$$p_{v_0 w} = \sigma^w(p_{v_0}) = \sum_{j=1}^m \beta_j \sigma_w(p_{v_j}) = \sum_{j=1}^m \beta_j p_{v_j w}, \quad (56)$$

which implies (51). \diamond

Step 2: Computation of a column basis J Having obtained the row generator $I \subseteq \Sigma^*$ in the step before, that is $\dim(X_t) \leq |I|$ and

$$\mathcal{R}(p) = \text{span}\{p_v \mid v \in I\}, \quad (57)$$

we can now use these functions p_v as an input for an algorithm which is analogous to that for computing the row generator I .

Algorithm 2

- 1: Define $q(w) := (p_v(w) = p(wv), v \in I) \in \mathbb{R}^{|I|}$.
- 2: $J \leftarrow \{\square\}, B_{col} \leftarrow \{q_w(\square)\}, C_{col} \leftarrow \Sigma$
- 3: **while** $C_{row} \neq \emptyset$ **do**
- 4: Choose $w \in C_{col}$.
- 5: **if** $q(w)$ is linearly independent of B_{col} **then**
- 6: $A_{col} \leftarrow A_{col} \cup \{w\}, B_{col} \leftarrow B_{col} \cup \{q(w)\}$
- 6: $C_{col} \leftarrow C_{col} \cup \{wa \mid a \in \Sigma\}$
- 7: **end if**
- 8: **end while**
- 9: **output** J

While this routine is, in essence, analogous to algorithm 1, there is one difference to be observed: Here C_{col} gets augmented by joining wa whereas C_{row} , in algorithm 1, was augmented by joining av . This asymmetry is due to that one obtains an equivalently asymmetric statement in lemma 2 when rephrasing it for $\mathcal{R}(p)$ instead of $\mathcal{C}(p)$. As a consequence, application of (34) instead of (35) in lemma 1 is needed.

We obtain that

$$P_{IJ} = [p(wv)]_{v \in I, w \in J} \quad (58)$$

is a generator for (X_t) . Since $\mathcal{R}(p) = \text{span}\{p_v \mid v \in I\}$, by applying (39), we see that

$$|J| = \dim(X_t). \quad (59)$$

Hence J is a genuine column basis. We recall that this was not necessarily the case for I which can happen to occur in the case $\mathcal{C}(p) \subsetneq \text{span}\{g_1, \dots, g_n\}$.

All $p(wav), v \in I, w \in J, a \in \Sigma$ can be obtained in runtime $O(|\Sigma| \cdot n^4)$ through application of (36) in lemma 1 making use of the $\vec{p}(w), \overleftarrow{p}(v)$ which were computed when executing the algorithms 1, 2.

We conclude: all necessary quantities can be obtained through $O(|\Sigma| \cdot n^4)$ arithmetic operations.

Step 3: Making I a basis This step is simple: one removes v from I where $p(wv), w \in J$ is linearly dependent in P_{IJ} . This reduces the possibly too large set I to a row basis and finally yields a basis (IJ) for (X_t) . This requires at most n linear independence tests of n -dimensional vectors hence $O(n^4)$ runtime [10].

5.2 Generating sets

Let us call a set $\{g_1, \dots, g_n\}$ of functions g_i as in the previous section a set of *generators* for the column space $\mathcal{C}(p)$ of the hidden Markov process resp. quantum random walk (X_t) .

We can get sets of generators as follows for which probabilities $g_i(v)$ can be computed in the style of hidden Markov processes resp. quantum random walks as follows.

Hidden Markov Processes Given a hidden Markov model $\mathcal{M} = (S, E, \pi, M)$, consider the hidden Markov models $\mathcal{M}_i = (S, X, \mathbf{e}_i, M)$, where \mathbf{e}_i is the i th unit vector in \mathbb{R}^S . One now takes

$$g_i(v) = \mathbf{e}_i^T T_v \mathbf{1} \quad (i = 1, \dots, n). \quad (60)$$

Quantum Random Walks For a quantum random walk, as parametrized through a self-adjoint matrix Q_{ψ_0} and linear operators $T_v, v \in \Sigma^*$ (acting on the state space see subsection 4.2), we see that

$$g_i(v) = \text{tr } T_v Q_i \quad (i = 1, \dots, n) \quad (61)$$

where the Q_i comprise all of the state space basis members $\mathbf{E}_{m_1 m_2}, \mathbf{F}_{m_1 m_2}$ (see (25)).

5.3 Summary

Theorem 2 yields the following procedure as an efficient test for equivalence of processes (X_t) and (Y_t) , :

1. Compute a basis for both (X_t) and (Y_t) .
2. If $\dim(X_t) \neq \dim(Y_t)$ return **not equivalent**.
3. If $\dim(X_t) = \dim(Y_t)$, perform equality tests from (13).
4. Output **equivalent** if all of them apply and **not equivalent** if not.

According to the above considerations, Step 1 can be performed in $O(|\Sigma|n^4)$ run-time where

$$n = \max\{n_X, n_Y\} \quad (62)$$

and n_X, n_Y , in case of hidden Markov processes $(X_t), (Y_t)$, are the numbers of hidden states and in case of quantum random walks $(X_t), (Y_t)$ are the dimensions of the associated state spaces. For step 2 we recall that all strings participating in the bases, as computed through algorithms 1,2, emerge as extensions of basis strings obtained in an earlier iterations. Application of (34,35,36) from lemma 1 then yields that all of the equality tests can be equally performed in $O(|\Sigma|n^4)$ arithmetic operations.

These insights can be condensed into the following main theorem where n as in (62).

Theorem 3. *The equivalence problem can be algorithmically solved for both hidden Markov processes and quantum random walks in $O(|\Sigma|n^4)$ arithmetic operations. \diamond*

Probabilistic Automata Our solution can be straightforwardly adapted to determine equivalence of probabilistic automata which we will describe in the following. It can therefore be viewed as more general than the main result obtained in [21]. The main difference one has to keep in mind is that probabilistic automata induce probability distributions on the (countable) set of strings Σ^* whereas HMMs give rise to stochastic processes, in other words to probability distributions on the (uncountably infinite) set of sequences $\Sigma^{\mathbb{N}}$. In case of probabilistic automata equivalence then translates to equality of the associated probability distributions on Σ^* . The following notations are adopted from [21].

Corollary 1. *Let $\mathcal{A}_1 = (S_1, \Sigma, M_1, \pi_1, F_1), \mathcal{A}_2 = (S_2, \Sigma, M_2, \pi_2, F_2)$ be two probabilistic automata where $N_1 = |S_1|, N_2 = |S_2|$. Then equivalence of $\mathcal{A}_1, \mathcal{A}_2$ can be determined in $O((|\Sigma| + 1)N^4)$ where $N = \max(N_1, N_2)$.*

Proof. By adding a special symbol $\$$ to Σ which is emitted from the final states with probability 1 the automata $\mathcal{A}_1, \mathcal{A}_2$ can be transformed into probabilistic automata with no final probabilities $\bar{\mathcal{A}}_1, \bar{\mathcal{A}}_2$. Let $p_{\bar{\mathcal{A}}_1}, p_{\bar{\mathcal{A}}_2}$ be the resulting stochastic processes. According to [5], lemmata 3 – 5, proposition 8, probabilistic automata with no final probabilities can be viewed as HMMs $\mathcal{M}_1, \mathcal{M}_2$ which translates to that for each $v \in \Sigma^*$

$$p_{\mathcal{A}_1}(v) = p_{\mathcal{M}_1}(v) \quad \text{and} \quad p_{\mathcal{A}_2}(v) = p_{\mathcal{M}_2}(v). \quad (63)$$

Note that the transformation from $\mathcal{A}_{1,2}$ to $\mathcal{M}_{1,2}$ requires only constant time. Applying theorem 1 to $\mathcal{M}_1, \mathcal{M}_2$ yields the result. \diamond

In short, corollary scales down the runtime $O((N_1 + N_2)^4)$ (the size of the alphabet $|\Sigma|$ is not discussed in [21]) to $O((\max(N_1, N_2))^4)$.

Ergodicity Tests In [20], a generic algorithmic strategy for testing ergodicity of hidden Markov processes was described, where overall efficiency hinged on computation of a basis of the tested hidden Markov processes. The algorithms described above resolve this issue. Hence ergodicity of hidden Markov processes can be efficiently tested. Similarly to the equivalence tests, the ergodicity test of [20] solely requires that the process in question is finitary. Therefore this efficient ergodicity test equally applies for quantum random walks.

5.4 Conclusive Remarks

We have presented a polynomial-time algorithm by which to efficiently test both hidden Markov processes and quantum random walks for equivalence. Previous solutions available for hidden Markov processes had runtime exponential in the number of hidden states. To test equivalence for quantum random walks, that is random walk models to be emulated on quantum computers, is relevant for the same reasons that apply for hidden Markov processes. An algorithm for testing equivalence for quantum random walks had not been available before. Note that the algorithm presented here is easy to implement and, in particular for hidden Markov processes, only requires invocation of well-known standard routines. Future directions are to explore how to efficiently test for *similarity* of hidden Markov processes and quantum random walks where similarity is measured in terms of *approximate equivalence*. Such tests have traditionally been of great practical interest.

References

1. D. Aharonov, A. Ambainis, J. Kempe, U. Vazirani: “Quantum walks on graphs”, *Proc. of 33rd ACM STOC, New York*, pp. 50–59, 2001.
2. A. Ambainis: “Quantum search algorithms”, *SIGACT News*, vol. 35(2), pp. 22–35, 2004.
3. D. Blackwell and L. Koopmans: “On the identifiability problem for functions of finite Markov chains”, *Annals of Mathematical Statistics*, vol. 28, pp. 1011–1015, 1957.
4. M.A. Nielsen and I.L. Chuang: “Quantum Computation and Quantum Information”, *Cambridge University Press*, Cambridge, UK, 2000.
5. P. Dupont, F. Denis and Y. Esposito: “Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms”, *Pattern Recognition*, vol. 38, pp. 1349–1371, 2005.
6. Durbin, Eddy, Krogh: “Biological Sequence Analysis” Cambridge University Press, 1998 (XXX: check)
7. Y. Ephraim and N. Merhav: “Hidden Markov Processes”, *IEEE Transactions on Information Theory*, vol. 48, 1518–1569, 2002.
8. U. Faigle and A. Schoenhuth: “Asymptotic mean stationarity of sources with finite evolution dimension”, *IEEE Transactions on Information Theory*, vol. 53, 2342–2348, 2007.
9. U. Faigle and A. Schoenhuth: “Discrete Quantum Markov Chains”, Preprint 2010, submitted.
10. D. K. Faddeev and V. N. Faddeeva: “Computational Methods of Linear Algebra”, *Freeman*, San Francisco, 1963.
11. E.J. Gilbert: “On the identifiability problem for functions of finite Markov chains”, *Annals of Mathematical Statistics*, vol. 30, pp. 688–697, 1959.

12. A. Heller: "On stochastic processes derived from Markov chains", *Annals of Mathematical Statistics*, vol. 36, pp. 1286–1291, 1965.
13. H. Ito, S.-I. Amari and K. Kobayashi: "Identifiability of hidden Markov information sources and their minimum degrees of freedom", *IEEE Trans. Inf. Theory*, vol. 38(2), pp. 324–333, 1992.
14. H. Jäger: "Observable operator models for discrete stochastic time series", *Neural Computation*, vol. 12(6), pp. 1371–1398, 2000.
15. J. Kempe: "Quantum random walks hit exponentially faster", *Probability Theory and Related Fields*, vol. 133(2), pp. 215–235, 2005.
16. F.L. Marquezino, R. Portugal, G. Abal and R. Donangelo: "Mixing times in quantum walks on the hypercube", *Phys. Rev. A*, 77, 042312, 2008.
17. Y. Ephraim, N. Merhav, "Hidden Markov processes", *IEEE Trans. on Information Theory*, vol. 48(6), pp. 1518–1569, 2002.
18. R. Pintelon, J. Schoukens, "System Identification", *IEEE Press*, Piscataway, NJ, 2001.
19. L.R. Rabiner: "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.
20. A. Schönhuth, H. Jaeger: "Characterization of ergodic hidden Markov sources", *IEEE Transactions on Information Theory*, vol. 55, pp. 2107–2118, 2009.
21. W.-G. Tzeng: "A polynomial-time algorithm for the equivalence of probabilistic automata", *SIAM Journal of Computing*, vol. 21, pp. 216–227, 1992.