Santa Fe Institute Working Paper 06-06-XXX
arxiv.org/quant-ph/0611203

Language Diversity of Measured Quantum Processes

Karoline Wiesner[1,2,*] and James P. Crutchfield[1,2,†]
[1] *Center for Computational Science & Engineering and Physics Department, University of California Davis, One Shields Avenue, Davis, CA 95616*
[2] *Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501*
(Dated: November 19, 2006)

The behavior of a quantum system depends on how it is measured. How much of what is observed comes from the structure of the quantum system itself and how much from the observer's choice of measurement? We explore these questions by analyzing the *language diversity* of quantum finite-state generators. One result is a new way to distinguish quantum devices from their classical (stochastic) counterparts. While the diversity of languages generated by these two computational classes is the same in the case of periodic processes, quantum systems generally generate a wider range of languages than classical systems.

PACS numbers: 03.67.Lx 03.67.-a 02.50.-r

I. INTRODUCTION

Quantum computation has advanced dramatically from Feynman's initial theoretical proposal [1] to the experimental realizations one finds today. The largest quantum device that has been implemented, though, is a 7 qubit register that can factor a 3 bit number [2] using Shor's algorithm [3]. A review of this and other currently feasible quantum devices reveals that, for now and the foreseeable future, they will remain small—in the sense that a very limited number of qubits can be stored. Far from implementing the theoretical ideal of a quantum Turing machine, current experiments test quantum computation at the level of small finite-state machines.

The diversity of quantum computing devices that lie between the extremes of finite-state and (unbounded memory) Turing machines is substantially less well understood than, say, that for classical automata, as codified in the Chomsky hierarchy [4]. As an approach to filling in a quantum hierarchy, comparisons between classical and quantum automata can be quite instructive.

Such results are found for automata at the level of finite-state machines [5–7]. For example, the regular languages are recognized by finite-state machines (by definition), but quantum finite-state machines, as defined in Ref. [6], cannot recognize all regular languages. This does not mean, however, that quantum automata are strictly less powerful than their classical counterparts. There are nonregular languages that are recognized by quantum finite-state machines [8]. These first results serve to illustrate the need for more work, if we are to fully appreciate the properties of quantum devices even at the lowest level of some presumed future quantum computational hierarchy.

The comparison of quantum and classical automata has recently been extended to the probabilistic languages recognized by stochastic and quantum finite-state machines [7]. There, quantum finite-state generators were introduced as models of the behaviors produced by quantum systems and as tools with which to quantify their information storage and processing capacities.

Here we continue the effort to quantify information processing in simple quantum automata. We will show how a quantum system's possible behaviors can be characterized by the diversity of languages it generates under different measurement protocols. We also show how this can be adapted to measurements, suitably defined, for classical automata. It turns out that the diversity of languages, under varying measurement protocols, provides a useful way to explore how classical and quantum devices differ. A measured quantum system and its associated measured classical system can generate rather different sets of stochastic languages. For periodic processes, the language diversities are the same between the quantum and counterpart classical systems. However, for aperiodic processes quantum systems are more diverse, in this sense, and potentially more capable.

In the following, we first review formal language and automata theory, including stochastic languages, stochastic and quantum finite-state generators, and the connection between languages and behavior. We then introduce the *language diversity* of a finite-state automaton and analyze a number of example processes, comparing quantum and classical models. We conclude with a few summary remarks and contrast the language diversity with *transient information*, which measures the amount of information an observer needs to extract in order to predict which internal state a process is in [9].

II. FORMAL LANGUAGES AND BEHAVIOR

Our use of formal language theory differs from most in how it analyzes the connection between a language and the systems that can generate it. In brief, we observe a

*Electronic address: `wiesner@cse.ucdavis.edu`
†Electronic address: `chaos@cse.ucdavis.edu`

system through a finite-resolution measuring instrument, representing each measurement with a *symbol* σ from discrete *alphabet* Σ. The temporal behavior of a system, then, is a string or a *word* consisting of a succession of measurement symbols. The collection of all (and only) those words is the *language* that captures the possible, temporal behaviors of the system.

Definition. *A* formal language $\mathcal{L}$ *is a set of* words $w = \sigma_0\sigma_1\sigma_2\ldots$ *each of which consists of a series of symbols* $\sigma_i \in \Sigma$ *from a discrete alphabet* Σ.

Σ^* denotes the set of all possible words of any length formed using symbols in Σ. We denote a word of length L by $\sigma^L = \sigma_0\sigma_1\ldots\sigma_{L-1}$, with $\sigma_i \in \Sigma$. The set of all words of length L is Σ^L.

Since a formal language, as we use the term, is a set of observed words generated by a process, then each *subword* $\sigma_i\sigma_{i+1}\ldots\sigma_{j-1}\sigma_j, i \leq j,\ i,j = 0,1,\ldots,L-1$ of a word σ^L has also been observed and is considered part of the language. This leads to the following definition.

Definition. *A language* $\mathcal{L}$ *is* subword closed *if, for each* $w \in \mathcal{L}$*, all of* w*'s subwords* $\mathrm{sub}(w)$ *are also members of* $\mathcal{L}$*:* $\mathrm{sub}(w) \subseteq \mathcal{L}$.

Beyond a formal language listing which words (or behaviors) occur and which do not, we are also interested in the probability of their occurrence. Let $\Pr(w)$ denote the probability of word w, then we have the following definition.

Definition. *A* stochastic language $\mathcal{L}$ *is a formal language with a* word distribution $\Pr(w)$ *that is normalized at each length* L*:*

$$\sum_{\{\sigma^L \in \mathcal{L}\}} \Pr(\sigma^L) = 1 \ , \tag{1}$$

with $0 \leq \Pr(\sigma^L) \leq 1$.

Definition. *Two stochastic languages* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ *are said to be* δ*-similar if* $\forall \sigma^L \in \mathcal{L}_1$ *and* $\sigma'^L \in \mathcal{L}_2 : |\Pr(\sigma^L) - \Pr(\sigma'^L)| \leq \delta$*, for all* L *and a specified* $0 \leq \delta \leq 1$*. If this is true for* $\delta = 0$*, then the languages are equivalent.*

For purposes of comparison between various computational models, it is helpful to refer directly to the set of words in a stochastic language $\mathcal{L}$. This is the *support* of a stochastic language:

$$\mathrm{supp}(\mathcal{L}) = \{w \in \mathcal{L} : \ \Pr(w) > 0\} \ . \tag{2}$$

The support itself is a formal language. Whenever we compare formal and stochastic languages we add the respective subscripts and write $\mathcal{L}_{formal}$ and $\mathcal{L}_{stoch}$.

III. STOCHASTIC FINITE-STATE GENERATORS

Automata with finite memory—*finite-state machines*—consist of a finite set of states and transitions between them [4]. Typically, they are used as *recognition* devices, whereas we are interested in the generation of words in a stochastic language. So here we will review models for classical and quantum generation, referring the reader to Ref. [10] for details on recognizers and automata in general.

Definition. *[7] A* stochastic generator G *is a tuple* $\{S, Y, \{T(y)\}\}$ *where*

1. *S is a finite set of states, with $|S|$ denoting its cardinality.*

2. *Y is a finite alphabet for output symbols.*

3. *$\{T(y), y \in Y\}$ is a set of $|Y|$ square stochastic matrices of order $|S|$. $|Y|$ is the cardinality of Y, the components $T_{ij}(y)$ give the probability of moving to state s_j and emitting y when in state s_i.*

4. *At each step a symbol $y \in Y$ is emitted and the machine updates its state. Thus, $\sum_{y \in Y} \sum_j T_{ij}(y) = 1$.*

Definition. *A* deterministic generator *(DG) is a* G *in which each matrix* $T(y)$ *has at most one nonzero entry per row.*

A. Process languages

Definition. *A* process language $\mathcal{P}$ *is a stochastic language that is subword closed.*

The output of a stochastic generator (as well as the quantum generator introduced below) is a process language; for the proof see Ref. [7]. Thus, all stochastic languages discussed in the following are process languages.

Definition. *A* periodic process language *with period* N *is a process language such that* $\forall w = \sigma_0\sigma_1\ldots\sigma_n \in \mathcal{P}$ *with* $n \geq N$*:* $\sigma_i = \sigma_{i+N}$.

Before discussing the languages associated with a G, we must introduce some helpful notation.

Notation. *Let* $|\eta\rangle = (11\ldots11)^T$ *denote a column vector with* $|S|$ *components that are all* 1*s.*

Notation. *The* state vector $\langle\pi| = (\pi_0, \pi_1, \ldots, \pi_{|S|-1})$ *is a row vector whose components,* $0 \leq \pi_i \leq 1$*, give the probability of being in state* s_i*. The state vector is normalized in probability:* $\sum_{i=0}^{|S|-1} \pi_i = 1$*. The* initial state distribution *is denoted* $\langle\pi^0|$.

The state-to-state transition probabilities of a G, independent of outputs, are given by the *state-to-state transition matrix*:

$$T = \sum_{y \in Y} T(y) \ , \tag{3}$$

which is a stochastic matrix: i.e., $0 \leq T_{ij} \leq 1$ and $\sum_j T_{ij} = 1$.

The generator updates its state distribution after each time step as follows:

$$\langle \pi^{t+1}| = \langle \pi^t | T(y) \ , \tag{4}$$

where (re)normalization of the state vector is assumed.

If a G starts in state distribution $\langle \pi^0|$, the probability of generating y^L is given by the state vector without renormalization

$$\Pr(y^L) = \langle \pi^0 | T(y^L) | \eta \rangle \ , \tag{5}$$

where $T(y^L) = \prod_{i=0}^{L-1} T(y_i)$ represents the assumption in our model that all states are accepting. This, in turn, is a consequence of our focusing on process languages, which are subword closed.

IV. QUANTUM GENERATORS

Quantum generators are a subset of *quantum machines* (or *transducers*), as defined in Ref. [7]. Their architecture consists of a set of internal states and transitions and an output alphabet that labels transitions. For simplicity here we focus on the definition of generators, without repeating the general definition of quantum transducers. Our basic *quantum generator* (*QG*) is defined as follows.

Definition. *[7] A QG is a tuple* $\{Q, \mathcal{H}, Y, \mathbf{T}(Y)\}$ *where*

1. $Q = \{q_i : i = 0, \ldots, n-1\}$ *is a set of* $n = |Q|$ internal states.
2. *The* state space $\mathcal{H}$ *is an* n*-dimensional Hilbert space.*
3. *The* state vector *is* $\langle \psi | \in \mathcal{H}$.
4. Y *is a finite alphabet for output symbols.* $\lambda \notin Y$ *denotes the null symbol.*
5. $\mathbf{T}(Y)$ *is a set of* n*-dimensional* transition matrices $\{T(y) = P(y) \cdot U, y \in Y\}$ *that are products of a unitary matrix* U *and a projection operator* $P(y)$ *where*
   (a) U *is an* n*-dimensional unitary* evolution operator *that governs the evolution of the state vector.*
   (b) $\mathbf{P}(Y)$ *is a set of* n*-dimensional* projection operators—$\mathbf{P} = \{P(y) : y \in Y \cup \{\lambda\}\}$—*that determines how a state vector is measured.The* $P(y)$ *are Hermitian matrices.*

At each time step a QG outputs a symbol $y \in Y$ *or the null symbol* λ *and updates its state vector.*

The output symbol y is identified with the measurement outcome. The symbol λ represents the event of no measurement. In the following we will concentrate on deterministic quantum generators. They are more transparent than general (nondeterministic) QGs, but still serve to illustrate the relative power of quantum and classical generators.

Definition. *A* quantum deterministic generator *(QDG) is a QG in which each matrix* $T(y)$ *has at most one nonzero entry per row.*

A. Observation and Operation

The projection operators determine how output symbols are generated from the internal, hidden dynamics. In fact, the only way to observe a quantum process is to apply a projection operator to the current state. In contrast with classical processes, the measurement event disturbs the internal dynamics. The projection operators are familiar from quantum mechanics and can be defined in terms of the internal states as follows.

Definition. *A* projection operator $P(y)$ *is the linear operator*

$$P(y) = \sum_{\kappa \in \mathcal{H}_y} |\phi_\kappa\rangle\langle\phi_\kappa| \ , \tag{6}$$

where κ *runs over the indices of a one- or higher-dimensional subspace* $\mathcal{H}_y$ *of the Hilbert space and the* ϕ_κ *span these subspaces.*

We can now describe a *QG*'s operation. U_{ij} is the transition amplitude from state q_i to state q_j. Starting in state $\langle \psi_0|$ the generator updates its state by applying the unitary matrix U. Then the state vector is projected using $P(y)$ and renormalized. Finally, symbol $y \in Y$ is emitted. In other words, a single time-step of a *QG* is given by:

$$\langle \psi(y)| = \langle \psi^0 | U P(y) \ , \tag{7}$$

where (re)normalization of the state vector is assumed. The state vector after L time steps when emitting string y^L is

$$\langle \psi(y^L)| = \langle \psi^0 | \prod_{i=0}^{L-1} \left(U P(y_i) \right) \ . \tag{8}$$

We can now calculate symbol and word probabilities of the process language generated by a *QG*. Starting the *QG* in $\langle \psi^0|$ the probability of output symbol y is given by the state vector without renormalization:

$$\Pr(y) = \left\| \psi(y) \right\|^2 \ . \tag{9}$$

By extension, the probability of output string y^L is

$$\Pr(y^L) = \left\| \psi(y^L) \right\|^2 \ . \tag{10}$$

B. Properties

In Ref. [7] we established a number of properties of QGs: their consistency with quantum mechanics, that they generate process languages, and their relation to stochastic generators and to quantum and stochastic recognizers. Here we avail ourselves of one property in particular of QDGs—for a given QDG there is always an equivalent (classical) deterministic generator. The latter is obtained by squaring the matrix elements of the QDG's unitary matrix and using the same projection operators. The resulting state-to-state transition matrix is doubly stochastic; i.e., $0 \leq T_{ij} \leq 1$ and $\sum_i T_{ij} = \sum_j T_{ij} = 1$.

Theorem 1. *Every process language generated by a QDG is generated by some DG.*

Proof. *See Ref. [7].*

This suggests that the process languages generated by QDGs are a subset of those generated by DGs. In the following, we will take a slightly different perspective and ask what set of languages a given QDG can generate as one varies the *measurement protocol*—that is, the choice of measurements.

V. LANGUAGE DIVERSITY

The notion of a measurement protocol is familiar from quantum mechanics: We define the *measurement period* as the number of applications of a projection operator relative to the unitary evolution time step. For a classical system this is less familiar, but it will be used in the same way. The measurement period here is the period of observing an output symbol relative to the internal state transitions. The internal dynamics remain unaltered in the classical case, whether the system is measured or not. In the quantum case, as is well known, the situation is quite different. Applying a projection operator disturbs the internal dynamics.

Definition. *A process observed with* measurement period p *is measured every* p *time steps.*

Note that this model of a measurement protocol, by which we subsample the output time series, is related to von Mises version of probability theory based on "collectives" [11].

The resulting observed behavior can be described in terms of the state-to-state transition matrix and the projection operators. For a classical finite-state machine this is:

$$\langle \pi(y)^{t+p}| = \langle \pi^t | T^{p-1} T(y) \ , \tag{11}$$

where $\langle \pi(y)^{t+p}|$ is the state distribution vector after p time steps and after observing symbol y. Note that $T(y) = TP(y)$.

For a quantum finite-state machine we have, instead:

$$\langle \psi(y)^{t+p}| = \langle \psi^t | U^p P(y) \ . \tag{12}$$

In both cases we dropped the renormalization factor.

The stochastic language generated by a particular quantum finite-state generator G for a particular measurement period p is labeled $\mathcal{L}^p(G)$. Consider now the set of languages generated by G for varying measurement period $\{\mathcal{L}^p(G)\}$.

Definition. *The* language diversity *of a (quantum or classical) finite-state machine G is the logarithm of the total number $|\{\mathcal{L}^p(G)\}|$ of stochastic languages that G generates as a function of measurement period p:*

$$\mathcal{D}(G) = \log_2 |\{\mathcal{L}^p(G)\}| \ . \tag{13}$$

Whenever we are interested in comparing the diversity in terms of formal and stochastic languages we add the respective subscript and write $\mathcal{D}_{formal}(G)$ and $\mathcal{D}_{stoch}(G)$, respectively. Here, $\mathcal{D}_{formal} = log_2 |\mathcal{L}^p_{formal}|$. In general, $\mathcal{D}_{stoch}(G) > \mathcal{D}_{formal}(G)$ for any particular G.

In the following we will demonstrate several properties related to the language diversity of classical and quantum finite-state machines.

Since every $\mathcal{L}(QDG)$ is generated by some DG, at first blush one might conclude that DGs are at least as powerful as QDGs. However, as pointed out in Ref. [7], this is true only for one particular measurement period. In the following examples we will study the dependence of the generated languages on the measurement period. It will become clear that Theorem 1 does not capture all of the properties of a QDG and its classical analog DG. For all but the periodic processes of the following examples the *language diversity* is larger for the QDG than its DG analog, even though the projection operators are identical.

These observations suggest the following.

Conjecture. $\mathcal{D}(QDG) \geq \mathcal{D}(DG)$.

The inequality becomes an equality in one case.

Proposition 1. *For a QDG G generating a periodic stochastic language $\mathcal{L}$ and its analog DG G'*

$$\mathcal{D}(G) = \mathcal{D}(G') \ . \tag{14}$$

Proof. *For any measurement period p and word length L words $y^L \in \mathcal{L}(G)$ and $y'^L \in \mathcal{L}(G')$ with $y^L = y'^L$ have the same probability:* $\Pr(y^L) = \Pr(y'^L)$. *That is,*

$$\Pr(y^L) = \|\psi^0 U^p P(y_0) U^p P(y_1) \ldots U^p P(y_{L-1})\|^2$$

and

$$\Pr(y'^L) = \langle \pi^0 | T^p P(y_0) T^p P(y_1) \ldots T^p P(y_{L-1}) | \eta \rangle \ .$$

Due to determinism and periodicity $\Pr(y^L) = 0$ *or* 1, *and also* $\Pr(y'^L) = 0$ *or* 1 *for all possible ψ^0 and π^0, respectively. Since $U = T$, the probabilities are equal.* □

We can give an upper bound for $\mathcal{D}$ in this case.

Proposition 2. *For a QG G generating a periodic process language $\mathcal{L}$ with period N:*

$$\mathcal{D}(G) \leq log_2(|Y| + N(N-1)) \ . \tag{15}$$

Proof. *Since $\mathcal{L}(G)$ is periodic, $\mathcal{L}^p(G) = \mathcal{L}^{p+N}(G)$. For $p = N, 2N, \ldots$: $\mathcal{L}^p(G) = \{y^*\}$, $y \in Y$. For $p = N + i, 2N + i, \ldots, 0 < i < N$: $\mathcal{L}^p(G) = \mathrm{sub}((\sigma_0\sigma_1 \ldots \sigma_{N-1})^*)$ and all its cyclic permutations are generated, in total N for each p. This establishes an upper bound of $|Y| + N(N-1)$.*

For general quantum processes there exists an upper bound for the language diversity.

Proposition 3. *For a QGD G*

$$\mathcal{D}(G) \leq log_2(|Y| + k(k-1)) \ , \tag{16}$$

where k is the integer giving

$$U^k = I + \iota J \ , \tag{17}$$

I is the identity matrix, $\iota \ll 1$, and J is a diagonal matrix $\sum_i |J_{ii}|^2 \leq 1$.

Proof. *It was shown in Ref. [6] (Thms. 6 and 7), that any $n \times n$ unitary U can be considered as rotating an $n-$dimensional torus. Then for some k U^k is within a small distance of the identity matrix. Thus, k can be considered the* pseudo-period *of the process, compared to a strictly periodic process with* period *N and $U^N = I$.*

Thus, $\mathcal{L}^p(G)$ and $\mathcal{L}^{p+k}(G)$ are δ-similar with $\delta \ll 1$. For $p = k$: $U^p = I + \iota J$, generating $\mathcal{L} = \{y^*\}$. Using the same argument as in the proof of Prop. 2 to lower the bound by k this establishes the upper bound for $\mathcal{D}(G)$.□

It should be noted that the upper bound on $\mathcal{D}$ depends on the parameter δ defining the similarity of languages $\mathcal{L}^p(G)$ and $\mathcal{L}^{p+k}(G)$. In general, the smaller δ is, the larger is k.

Proposition 4. *For a QDG G generating a periodic process language the number of formal languages $|\mathcal{L}_{formal}(G)|$ equals the number of stochastic languages $|\mathcal{L}_{stoch}(G)|$*

$$\mathcal{D}_{formal}(G) = \mathcal{D}_{stoch}(G). \tag{18}$$

Proof. *It is easily seen that any QG generating a periodic process is deterministic: its unitary matrix has only 0 and 1 entries. It follows that word probabilities are either 0 or 1 and so there is a one-to-one mapping between the stochastic language generated and the corresponding formal language.*□

Corollary 1. *For a QDG G generating a periodic process and its analog DG G′:*

$$\mathcal{D}_{formal}(G) = \mathcal{D}_{formal}(G') = \mathcal{D}_{stoch}(G) = \mathcal{D}_{stoch}(G') \ . \tag{19}$$

Proof. *The Corollary follows from Prop. 1 and a straightforward extension of Proposition 4 to classical periodic processes.*□

VI. EXAMPLES

The first two examples, the iterated beam splitter and the quantum kicked top, are quantum dynamical systems that are observed using complete measurements. In quantum mechanics, a *complete measurement* is defined as a nondegenerate measurement operator, i.e., one with nondegenerate eigenvalues. The third example, the distinct period-5 processes, illustrates processes observed via incomplete measurements. Deterministic quantum and stochastic finite-state generators are constructed and compared for each example.

A. Iterated beam splitter

The iterated beam splitter is a simple quantum process, consisting of a photon that repeatedly passes through a loop of beam splitters and detectors, with one detector between each pair of beam splitters [7]. Thus, as the photon traverses between one beam splitter and the next, its location in the upper or lower path between them is measured nondestructively by the detectors. The resulting output sequence consists of symbols 0 (upper path) and 1 (lower path).

The operators have the following matrix representation in the experiment's eigenbasis:

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \ , \tag{20}$$

$$P(0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \ , \tag{21}$$

$$P(1) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \ . \tag{22}$$

Observing with different measurement periods, the generated language varies substantially. As can be easily seen with Eqs. (10) and (12), three (and only three) languages are generated as one varies p. They are summarized in Table I for all $y^L \in \mathcal{L}$ and for $n = 0, 1, 2 \ldots$, which is used to parametrize the measurement period. The language diversity of the QDG is then $\mathcal{D} = \log_2(3)$. We can compare this to the upper bound given in Prop. 3. In the case of the unitary matrix U given above $k = 2$, since $UU = I$. U is also known as the *Hadamard* matrix. Thus, the upper bound for the language diversity in this case is $\mathcal{D} \leq log_2(4)$.

The classical equivalent DG for the iterated beam splitter, constructed as described in Ref. [7], is given by the following state-to-state transition matrix:

$$T = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \ .$$

Using Eqs. (5) and (11), we see that only one language is generated for all p. This is the language of the *fair coin* process, a random sequence of 0s and 1s, see Table I. Thus, $\mathcal{D}(DG) = 0$.

Iterated Beam Splitter Language Diversity				
Machine Type	p	supp($\mathcal{L}$)	$\mathcal{L}$	$\mathcal{D}$
QDG	$2n$	$(0+1)^*$	$\Pr(y^L) = 2^{-L}$	
	$2n+1$	0^*	$\Pr(y^L) = 1$	
	$2n+1$	1^*	$\Pr(y^L) = 1$	1.58
DG	n	$(0+1)^*$	$\Pr(y^L) = 2^{-L}$	0

TABLE I: Process languages generated by the QDG for the iterated beam splitter and by the classical DG. The measurement period takes a parameter $n = 0, 1, 2 \ldots$. The word probability is given for all $y^L \in \mathcal{L}$.

B. Quantum kicked top

The periodically kicked top is a familiar example of a finite-dimensional quantum system whose classical limit exhibits various degrees of chaotic behavior as a function of its control parameters [12]. For a spin-1/2 system the unitary matrix is:

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} e^{-ik} & 0 \\ 0 & e^{-ik} \end{pmatrix}$$

and the projection operators are:

$$P(0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} ,$$

$$P(1) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} .$$

Since this QDG G is deterministic, its classical DG G' exists and is given by:

$$T = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} .$$

The process languages generated by this QDG and its analog DG are given in Table II. The language diversity is $\mathcal{D}(G) = \log_2(5)$. Whereas the language diversity of classical counterpart DG is $\mathcal{D}(G') = 0$, since it generates only the language of the fair coin process.

C. Period-5 process

As examples of periodic behavior and, in particular, of incomplete measurements, consider the binary period-5 processes distinct up to permutations and $(0 \leftrightarrow 1)$ exchange. There are only three such processes: $(11000)^*$, $(10101)^*$, and $(10000)^*$ [13]. They all have the same state-to-state transition matrix—a period-5 permutation. This irreducible, doubly stochastic matrix is responsible for the fact that the QDG of a periodic process and its classical DG have the same properties. Their

Spin-1/2 Quantum Kicked Top Language Diversity				
Machine Type	p	supp($\mathcal{L}$)	$\mathcal{L}$	$\mathcal{D}$
QDG	$4n+1, 4n+3$	$(0+1)^*$	$\Pr(y^L) = 2^{-L}$	
	$4n+2$	sub($(01)^*$)	$\Pr(((01)^*)^L) = 1/2$	
			$\Pr(((10)^*)^L) = 1/2$	
	$4n+2$	sub($(10)^*$)	$\Pr(((10)^*)^L) = 1/2$	
			$\Pr(((01)^*)^L) = 1/2$	
	$4n$	0^*	$\Pr(y^L) = 1$	
	$4n$	1^*	$\Pr(y^L) = 1$	2.32
DG	n	$(0+1)^*$	$\Pr(y^L) = 2^{-L}$	0

TABLE II: Process languages generated by the QDG for the spin-1/2 quantum kicked top and its corresponding classical DG. The measurement period, again, is parametrized by $n = 0, 1, 2 \ldots$. The word probability is given for all $y^L \in \mathcal{L}$.

state-to-state unitary transition matrix is given by

$$T = U = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} . \tag{23}$$

The projection operators differ between the processes with different template words, of course. For template word 10000, they are:

$$P(0) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} , \tag{24}$$

$$P(1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} . \tag{25}$$

For 11000, they are:

$$P(0) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} , \tag{26}$$

$$P(1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} . \tag{27}$$

Distinct Period-5 Processes' Language Diversity				
Machine Type	p	$\mathrm{supp}(\mathcal{L})$	$\mathcal{L}, L > 5$	$\mathcal{D}$
10000	$5n+1, 5n+2$	$\mathrm{sub}((10000)^*)$	$\Pr(y^L) = 1/5$	
	$5n+3, 5n+4$			
	$5n$	0^*	$\Pr(y^L) = 1$	
	$5n$	1^*	$\Pr(y^L) = 1$	1.58
11000	$5n+1, 5n+4$	$\mathrm{sub}((11000)^*)$	$\Pr(y^L) = 1/5$	
	$5n+2, 5n+3$	$\mathrm{sub}((01010)^*)$	$\Pr(y^L) = 1/5$	
	$5n$	0^*	$\Pr(y^L) = 1$	
	$5n$	1^*	$\Pr(y^L) = 1$	2
10101	$5n+1, 5n+4$	$\mathrm{sub}((10101)^*)$	$\Pr(y^L) = 1/5$	
	$5n+2, 5n+3$	$\mathrm{sub}((00111)^*)$	$\Pr(y^L) = 1/5$	
	$5n$	0^*	$\Pr(y^L) = 1$	
	$5n$	1^*	$\Pr(y^L) = 1$	2

TABLE III: Process languages produced by the three distinct period-5 generators. The quantum and classical versions are identical in each case. The measurement period is parametrized by $n = 0, 1, 2 \ldots$. For simplicity, the word probability is given for all $y^L \in \mathcal{L}$ with $L \geq 5$. For the nontrivial languages above, when $L > 5$ there are only five words at each length, each having equal probability.

	Quantum process	Classical process
System	Iterated beam splitter	Fair coin
$\mathcal{D}$	$\log_2(3)$	0
Measurement	Complete	Complete
System	Quantum kicked top	Fair coin
$\mathcal{D}$	$log_2(5)$	0
Measurement	Complete	Complete
System	10000	10000
$\mathcal{D}$	$\log_2(3)$	$\log_2(3)$
Measurement	Incomplete	Incomplete
System	11000	11000
$\mathcal{D}$	2	2
Measurement	Incomplete	Incomplete
System	10101	10101
$\mathcal{D}$	2	2
Measurement	Incomplete	Incomplete

TABLE IV: Comparison between QDGs and their classical DGs. Note that the term "(in)complete measurement" is not used for classical systems. However, the above formalism does render it meaningful. It is used in the same way as in the quantum case (one-dimensional subspaces or non-degenerate eigenvalues).

And for word 10101, they are:

$$P(0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} , \tag{28}$$

$$P(1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} . \tag{29}$$

The difference between the measurement alphabet size and the period of a process, which determines the number of states of a periodic process, should be noted. In all our examples the measurement alphabet is binary. Thus, in having five internal states but only a two-letter measurement alphabet, the period-5 processes necessarily constitute systems observed via incomplete measurements.

The set of languages generated by the three processes is summarized in Table III. The generated language depends on the initial state only when the measurement period is a multiple of the process period.

The language diversity for the process 10000 is $\mathcal{D} = \log_2(3)$ and for both the processes 11000 and 10101, $\mathcal{D} = 2$. Note that the processes 11000 and 10101 generate each other at particular measurement periods, if one exchanges 0s and 1s. It is not surprising therefore that the two models have the same language diversity.

It turns out that the state of the quantum systems under periodic dynamics is independent of the measurement protocol. At each point in time the system is in an eigenstate of the measurement operator. Therefore, the measurement does not alter the internal state of the quantum system. Thus, a system in state $\langle\psi_0|$ is going to be in a particular state $\langle\psi_2|$ after two time steps, independent of whether being measured in between. This is true for quantum and classical periodic systems. The conclusion is that for periodic processes there is no difference between unmeasured quantum and classical states. This is worth noting, since this is the circumstance where classical and quantum systems are supposed to differ. As a consequence the language diversity is the same for the quantum and classical model of all periodic processes, which coincides with Prop. 1.

Note, however, that the language diversity is not the same for all processes with the same period. A property that is reminiscent of the transient information [9, 13], which also distinguishes between structurally different periodic processes.

D. Discussion

The examples show that the language diversity monitors aspects of a process's structure and it is different for quantum and classical models of aperiodic processes. This suggests that it will be a useful aid in discovering structure in the behavior of quantum dynamical systems. For the aperiodic examples, the QDG had a larger language diversity than its classical DG. And this suggests a kind of computational power of QDGs that is not obvious from the structural constraints of the machines. Language diversity could be compensation, though, for other limitations of QDGs, such as not being able to generate all regular languages. The practical consequences of this for designing quantum devices remains to be explored.

A comparison between QDGs and their classical DGs gives a first hint at the structure of the lowest levels of a potential hierarchy of quantum computational model classes. It turned out that for periodic processes a QDG has no advantage over a DG in terms of the diversity of languages possibly generated by any QDG. However, for the above examples of both incomplete and complete measurements, the set of generated stochastic languages is larger for a QDG than the corresponding DG.

Table IV summarizes the processes discussed above, their properties and language diversities. All finite-state machines are deterministic, for which case it was shown that there exists an equivalent DG that generates the same language [7]. This is true, though only for one particular measurement period. Here we expanded on those results in comparing a range of measurement periods and the entire set of generated stochastic languages.

For each example quantum generator and the corresponding classical generator the language diversity and the type of measurement (complete/incomplete) are given. For all examples the language diversity is larger for the QDG than the DG. It should be noted, however, that the fair coin process is also generated by a one-state DG with transition matrices $T(0) = T(1) = (1/2)$. This it not true for the QDGs. Thus, the higher language diversity of a QDG is obtained at some cost—a larger number of states is needed than with a DG generating any one particular process language. The situation is different, again, for the period-5 processes—there is no DG with fewer states that generates the same process language.

The above examples were simple in the sense that their language diversity is a finite, small number. In some broader sense, this means that they are recurrent—to use terminology from quantum mechanics. For other processes the situation might not be quite as straightforward. To find the language diversity one has to take the limit of large measurement periods. For implementations this is a trade-off, since larger measurement period requires a coherent state for a longer time interval. In particular it should be noted that in the above examples shorter intervals between measurements cause more "interesting" observed behavior. That is, the stochastic language $\mathcal{L}^2 = \{(01)^*, (10)^*\}$ generated by the quantum kicked top with $\Pr(y^L) = 1/2$, consisting of strings with alternating 0s and 1s is more structured than the language $\mathcal{L}^4 = \{0^*\}$ with $\Pr(y^L) = 1$ consisting of only 0s. (Cf. Table II.)

VII. CONCLUSION

Quantum finite-state machines occupy the lowest level of an as-yet only partially known hierarchy of quantum computation. Nonetheless, they are useful models for quantum systems that current experiment can implement, given the present state of the art. We briefly reviewed quantum finite-state generators and their classical counterparts—stochastic finite-state generators. Illustrating our view of computation as an intrinsic property of a dynamical system, we showed similarities and differences between finite-memory classical and quantum processes and, more generally, their computational model classes. In particular, we introduced the language diversity—a new property that goes beyond the usual comparison of classical and quantum machines. It captures the fact that, when varying measurement protocols, different languages are generated by quantum systems. Language diversity appears when quantum interference operates.

For a set of examples we showed that a deterministic quantum finite-state generator has a larger language diversity than its classical analog. Since we associate a language with a particular behavior, we also associate a set of languages with a set of possible behaviors. As a consequence, the QDGs all exhibited a larger set of behaviors than their classical analogs. That is, they have a larger capacity to store and process information.

We close by suggesting that the design of finite quantum computational elements could benefit from considering the measurement process not only as a final but also as an intermediate step, which may simplify experimental design.

Since we considered only finite-memory systems here, their implementation is already feasible with current technology. Cascading compositions of finite processes can rapidly lead to quite sophisticated behaviors, as discussed in Ref. [7]. A discussion of associated information storage and processing capacity analogous to those used for classical dynamical systems in Ref. [9] is under way.

Acknowledgments

Partial support was provided by DARPA Agreement F30602-00-2-0583. KW's postdoctoral fellowship was provided by the Wenner-Gren Foundations, Stockholm, Sweden.

[1] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.

[2] E. Knill, R. Laflamme, R. Martinez, and C.-H. Tseng. An

algorithmic benchmark for quantum information processing. *Nature*, 405:368–370, 2002.

[3] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, page 124, 1994. e-print arxiv/quant-ph/9508027.

[4] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2001.

[5] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *38th IEEE Conference on Foundations of Computer Science*, pages 66–75, 1997.

[6] C. Moore and J. P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, 237:275–306, 2000.

[7] K. Wiesner and J. P. Crutchfield. Computation in finitary quantum processes. *in preparation*, 2006.

[8] A. Bertoni and M. Carpentieri. Analogies and differences between quantum and stochastic automata. *Theoretical Computer Science*, 262:69–81, 2001.

[9] J. P. Crutchfield and D. P. Feldman. Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos*, 13:25 – 54, 2003.

[10] A. Paz. *Introduction to Probabilistic Automata*. New York Academic Press, 1971.

[11] C. Howson. Theories of probability. *The British Journal for the Philosophy of Science*, 46:1–32, 1995.

[12] F. Haake, M. Kús, and R. Scharf. Classical and quantum chaos for a kicked top. *Z. Phys. B*, 65:381–395, 1987.

[13] D. P. Feldman and J. P. Crutchfield. Synchronizing to periodicity: The transient information and synchronization time of periodic sequences. *Advances in Complex Systems*, 7(3-4):329–355, 2004.