# The values of rational probabilistic finite automata are the left-c.e. reals

Seth Douglas Nidhal Mghirbi

Version 2.0.0, November 2026

**Abstract**

The values of probabilistic finite automata with rational transitions and rational initial distributions are exactly the left-c.e. reals in $[0, 1]$. Here value means the supremum of acceptance probabilities over finite words. Two letters and one fixed number of states suffice; in particular a single automaton can have a noncomputable value, and every prefix-free halting probability occurs. We prove the same spectrum for one-way Kondacs–Watrous measure-many quantum automata, already with two real rational orthogonal gates, a basis start and one fixed dimension. We also construct one fixed probabilistic automaton and one fixed quantum apparatus whose spectra are full when only the rational start varies. For these Kondacs–Watrous automata, value one, nonisolation at $1/2$, and density of the word probabilities are $\Pi^0_2$-complete; the two isolation problems of Derksen, Jeandel and Koiran are $\Sigma^0_2$-complete. For our two fixed apparatuses, exact preparation selectors with ordinary program indices as input have optimal degree $0'$ on interior targets; every fixed probabilistic automaton with full spectrum requires $0''$ on $[0, 1]$, whereas a modified fixed quantum apparatus needs only $0'$ there. The quantum construction uses graph projections to charge errors in reported computations while a live reservoir permits repeated trials. Measure-once values with algebraic data remain algebraic. Latvian value computability and isolation remain open.

AI systems assisted with this work; see "How this work was done."

## Contents

**1 Introduction** **2**
1.1 How the automaton works . . . 4
1.2 Prior work . . . 4

**2 Preliminaries** **5**

**3 Counter programs that output words** **6**

**4 Checking a run with two random tests** **9**
4.1 The equality test . . . 9
4.2 Certificates . . . 9
4.3 The parser . . . 10
4.4 The checking automaton . . . 10

**5 The payout and the proof of Theorem A** **12**

**6 An explicit noncomputable value and halting probabilities** **13**

**7 Two letters and a fixed number of states** **15**

**8 Uniformity** **16**

**9 Quantum automata: where the ladder breaks** **17**
9.1 The two ends . . . 17
9.2 Quantum automata with control language . . . 18
9.3 The one-way measure-many model . . . 19
9.4 The Latvian boundary . . . 22

**10 Open problems** **23**

**11 The one-way measure-many spectrum** **25**
11.1 A projection compiler . . . 25
11.2 A finite graph which checks an emitting computation . . . 29
11.3 Every left-c.e. real, with fixed dimension . . . 31

**12 Full spectra with fixed hardware** **32**
12.1 Addresses and bins . . . 32
12.2 One fixed probabilistic automaton . . . 33
12.3 One fixed quantum apparatus . . . 34

**13 Isolation, density and arithmetic index sets** **35**
13.1 Arithmetic of the optimum . . . 37

**14 The degrees of exact preparation selectors** **38**

**15 A quantum selector on the closed interval** **40**

**How this work was done** **43**

**Verification status** **45**

**Credit** **46**

# 1 Introduction

A probabilistic finite automaton, introduced by Rabin [26], has finitely many states. On each input letter it moves from its current state to a random next state, with probabilities given by a stochastic matrix for that letter, and it accepts a word if it ends in an accepting state. We consider automata whose transition probabilities and initial distribution are rational. The *value* $V(A)$ of such an automaton $A$ is the supremum, over all finite words $w$, of the probability $P_A(w)$ that $A$ accepts $w$. Which real numbers are values?

A real number is *left-c.e.* if it is the limit of a computable nondecreasing sequence of rational numbers [10]. Every value is left-c.e.: each $P_A(w)$ is a rational number computable from $A$ and $w$, and the maximum of $P_A$ over the first $n$ words of an enumeration increases to $V(A)$ (Proposition 2.3). This observation is folklore; it amounts to the semi-decidability of the strict emptiness problem, whether $V(A) > \lambda$ for a given rational $\lambda$. Our main result is the converse.

*Theorem A.* The values of rational probabilistic finite automata are exactly the left-c.e. reals in $[0, 1]$.

*Theorem B.* There is a number $N$ such that every left-c.e. real in $[0, 1]$ is the value of a rational probabilistic finite automaton with two input letters and at most $N$ states.

The number $N$ is fixed by one universal counter program (Section 7); we have not computed it, and it is large. Two states do not suffice: an automaton with two states has a rational value (Proposition 10.1).

It has long been known that the value cannot be computed from the automaton. Strict emptiness is undecidable [24, 25]; the value cannot be approximated to within any fixed error smaller than $1/2$ [8, 20]; and whether $V(A) = 1$ is undecidable [15]. Fijalkow derives these results from one theorem [12]. They are statements about the map $A \mapsto V(A)$, and they are compatible with every single value being a computable real, in the same way that equality to 0 is undecidable for computable reals although each real in question is computable. The remark of Fijalkow, Gimbert, Kelmendi and Oualhadj that, as a consequence of Paz's undecidability result, "the value of an automaton is not computable in general" [13] refers to this map. Rote proved that strict emptiness is undecidable even for one fixed automaton whose initial distribution, or whose vector of final weights, is the input [27, 28]; the input there is not the threshold, and his theorem does not show that the value of one automaton is noncomputable. For channels, Boche, Schaefer and Poor treat a single computable finite-state channel with a noncomputable capacity as a stronger, hypothetical form of non-computability, and cite examples for compound and averaged channels with countably infinite sets of states [5, Remark 31]. For probabilistic finite automata, Theorem C below gives one automaton with a noncomputable value, and Theorem A shows that every left-c.e. real occurs; through the channels of Elkouss and Pérez-García [11], the same holds for capacities of finite-state channels with rational transition probabilities (Corollary 10.7).

*Theorem C.* Fix an effective enumeration of Turing machines. Let $H$ be the set of numbers $n$ such that the $n$-th machine halts on input $n$, and let $\alpha_* = 1/4 + \sum_{n \in H} 4^{-(n+2)}$. Then $\alpha_*$ is not computable, and it is the value of the rational probabilistic finite automaton $A_*$ of Section 6, built with the constants $a = 1/8$, $b = 1/2$ and $B = 16$.

In particular, for the one automaton $A_*$, the strict emptiness problem with the threshold as the only input is undecidable. Once an effective enumeration of Turing machines is fixed, the automaton $A_*$ is obtained from a specified Turing machine by the compilation of Sections 3 to 5; its matrices are large, and we do not list them. Reals defined in this way from a computably enumerable undecidable set go back to Specker [30].

*Corollary D.* The halting probability of every prefix-free machine, in particular Chaitin's halting probability $\Omega$ of a universal prefix-free machine, is the value of a rational probabilistic finite automaton with two input letters and at most $N$ states.

It is well known that every left-c.e. real in $[0, 1]$ is the halting probability of some prefix-free machine [6, 10], so Theorem A says that the values of rational probabilistic finite automata are exactly the halting probabilities of prefix-free machines (Remark 6.5).

The construction is effective once a rational interval $a < \alpha < b$ around the target is given, and it cannot be made effective from a left-c.e. index of the target alone: whether $V(A) = 0$ can be decided from $A$, while whether a left-c.e. real given by an approximating sequence is positive cannot (Proposition 8.1).

Theorem A has the shape of the theorem of Hochman and Meyerovitch that the topological entropies of shifts of finite type on $\mathbb{Z}^d$, $d \geq 2$, are exactly the nonnegative right-c.e. reals [16]. In both cases a quantity attached to a finite description has an evident one-sided approximation, from above for entropy and from below for the value of an automaton, and every real with such an approximation occurs. We use this only as an analogy; the methods are different.

For quantum automata the measure-once end has algebraic values for algebraic data, whereas one-way Kondacs–Watrous measure-many automata already have the full left-c.e. spectrum. The latter assertion, including fixed dimension, rational orthogonal gates and a binary alphabet, is proved in Theorem 11.5. Thus both questions left open in version 1.1.0 have negative answers. The fixed-matrix question of that version also has a positive answer:

one fixed PFA, and one fixed quantum apparatus with rational pure starts, each have the full spectrum (Theorems 12.2 and 12.4). The value-one and isolation decisions have exact arithmetic classifications (Theorem 13.2). Selection of exact rational starts has an additional representation issue: our oracle-degree theorems take ordinary program indices, not streams of approximants (Section 14).

### 1.1 How the automaton works

Fix a left-c.e. real $\alpha \in (0,1)$ and rationals $0 < a < \alpha < b < 1$. For a binary word $s = s_1 \cdots s_k$ let $x(s) = \sum_{i=1}^{k} s_i 2^{i-k-1}$, the binary fraction $0.s_k s_{k-1} \cdots s_1$, and let $q(s) = a + (b-a)x(s)$. The set $W_\alpha$ of words $s$ with $q(s) < \alpha$ is computably enumerable, so some counter program outputs exactly the words of $W_\alpha$ along its halting runs (Section 3). The input of the automaton is meant to be a sequence of *certificates*, each listing a run of this program: its instructions in order, and the counter values in unary (Figure 1). A deterministic part of the automaton checks everything except the counter arithmetic. The arithmetic, which a finite automaton cannot check, is tested at random. At the start of each certificate the automaton picks one of two tests with equal probability. The *joint* test survives each unary symbol of both sides of every claimed equation with probability $1/B$; the *split* test picks one side of each equation at random and survives each of its symbols with probability $1/B^2$. On a correct run the two tests survive with the same probability. If any equation is false, the split test is at least $\kappa_B = (B + B^{-1})/2$ times as likely to survive (Lemma 4.1 and Corollary 4.6). Independently, a two-state register holds $x(s)$, for the output $s$ of the claimed run, as the probability of one of its states. At the end of a certificate, a surviving joint test accepts with probability $q(s)$, a surviving split test rejects with probability $1 - q(s)$, and all other probability starts afresh. Acceptance and rejection are permanent.

Of the probability that a certificate settles, a correct certificate therefore accepts the fraction $q(s) < \alpha$, and an incorrect one a fraction at most $b/(b + \kappa_B(1-b))$, which is below $a$ once $B$ is large. Every input word, however it mixes correct, incorrect and malformed certificates, is accepted with probability at most $\alpha$. Repeating one correct certificate drives the acceptance probability up to $q(s)$, and these rewards approach $\alpha$.

### 1.2 Prior work

Table 1 lists what was known about the value and what is proved here. The verification of counter-machine runs by a probabilistic finite automaton, through coin-flipping comparisons of unary lengths, is due to Condon and Lipton [8], who used an equality test of Freivalds [14] to prove that emptiness is undecidable with a gap; Madani, Hanks and Condon [19, 20] and Rote [27] give detailed accounts. Our joint and split tests are a form of this test. What we add is the payout at the delimiter: the surviving tests accept and reject in proportions set by a reward read from the certificate's output, so that the value of the automaton is the supremum of the rewards rather than one of two values separated by a gap. For partially observable Markov decision processes the value is likewise known to be uncomputable as a function of the process [20]; Proposition 10.5 records the range of values for reachability objectives.

**Reading guide.** Section 2 fixes definitions and proves the easy inclusion. Section 3 constructs counter programs that output a given computably enumerable set of words; it is Minsky's simulation of Turing machines with outputs added, and a reader who knows it can take Lemmas 3.2 and 3.3 as given. Section 4 builds the random check of one certificate, and Section 5

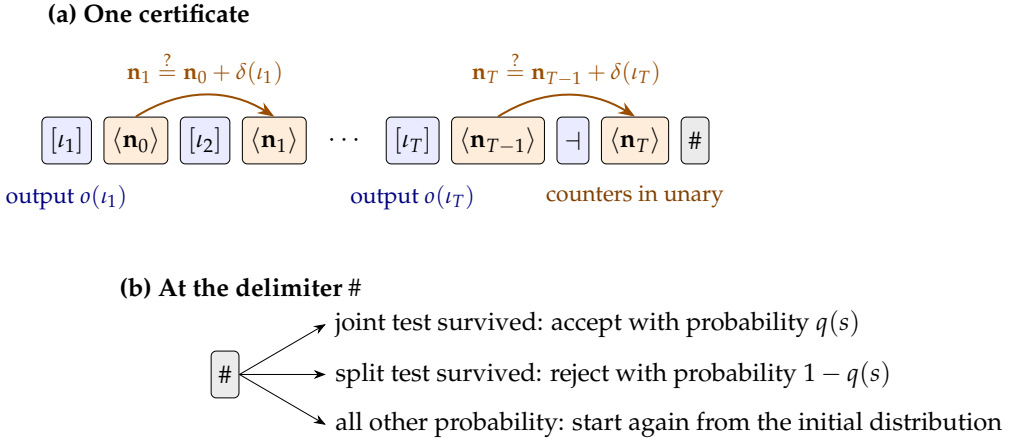


**Figure 1:** (a) A certificate lists a claimed run of a counter program. Each row is the next instruction, or $\dashv$ for the last row, followed by the counter values in unary. A deterministic parser checks everything except the equations between consecutive rows, which are tested at random. (b) At the delimiter a surviving joint test accepts with probability $q(s)$ and a surviving split test rejects with probability $1 - q(s)$, where $s$ is the output listed in the certificate; everything else starts again. Acceptance and rejection are permanent.

adds the payout and proves Theorem A. Section 6 proves Theorem C and Corollary D, the latter using Theorem B, which Section 7 proves; and Section 8 discusses uniformity. Section 9 compares quantum automata, and Section 10 collects further open problems, with two classes of automata whose values are computable and the corresponding statement for partially observable Markov decision processes.

The additions in version 2.0.0 follow the retained sections: Section 11 proves the quantum range theorem, Section 12 fixes the hardware, Section 13 treats isolation and density, Section 14 proves the selector degrees, and Section 15 gives a fixed quantum apparatus with a selector of degree $0'$ on the closed interval. All values throughout are finite-word suprema; eventual acceptance on a prescribed infinite stream is a different notion and is not used here.

## 2 Preliminaries

**Definition 2.1.** A *rational probabilistic finite automaton* is a tuple $A = (Q, \Sigma, (M_\ell)_{\ell \in \Sigma}, \pi, F)$, where $Q$ and $\Sigma$ are finite nonempty sets, each $M_\ell$ is a row-stochastic $Q \times Q$ matrix with rational entries, $\pi$ is a probability row vector on $Q$ with rational entries, and $F \subseteq Q$. For a word $w = \ell_1 \cdots \ell_k$ over $\Sigma$ let $M_w = M_{\ell_1} \cdots M_{\ell_k}$, with $M_\varepsilon$ the identity, and let $P_A(w) = \pi M_w \mathbf{1}_F$, where $\mathbf{1}_F$ is the indicator column of $F$. The *value* of $A$ is $V(A) = \sup_{w \in \Sigma^*} P_A(w)$.

The empty word is included, so $V(A) \geq \pi \mathbf{1}_F$. We describe automata by random transition rules: to say that on letter $\ell$ the automaton moves from state $p$ to a random state with a given law means that row $p$ of $M_\ell$ is that law. An automaton made of finitely many components, each with finitely many states, that update by independent random choices has the product of the component state sets as its state set, and the entries of its matrices are products of the probabilities of the components; it is a rational automaton when all these probabilities are rational.

**Definition 2.2.** A real $\alpha$ is *left-c.e.* if $\alpha = \lim_n \alpha_n$ for a computable nondecreasing sequence $(\alpha_n)$ of rationals. An index of a Turing machine that computes $n \mapsto \alpha_n$ is a *left-c.e. index* of $\alpha$.

| Statement | Scope | Source |
|---|---|---|
| Every value is left-c.e. | every automaton | folklore |
| Whether $V(A) > \lambda$ is undecidable | uniform in $A, \lambda$ | [24, 25] |
| $V(A)$ cannot be approximated to within any fixed error below $1/2$ | uniform in $A$ | [8, 20, 12] |
| Whether $V(A) = 1$ is undecidable | uniform in $A$ | [15] |
| Emptiness is undecidable for one fixed set of matrices, with the initial distribution or the final vector as input | one set of matrices, varying vectors | [27, 28] |
| One automaton has a noncomputable value | one automaton | Theorem C |
| The values are exactly the left-c.e. reals in $[0, 1]$ | every left-c.e. real | Theorem A |
| Two letters and at most $N$ states realize every value | one bound for all values | Theorem B |
| Every prefix-free halting probability is a value | every halting probability | Corollary D |

**Table 1:** What was known about the value $V(A)$ of a rational probabilistic finite automaton, and what is proved here.

If $(\alpha_n)$ is such a sequence, a rational $r$ satisfies $r < \alpha$ if and only if $\alpha_n > r$ for some $n$. Hence the set of rationals below $\alpha$ is computably enumerable, uniformly in a left-c.e. index of $\alpha$. An *enumeration index* of a computably enumerable set $W$ is an index, in a fixed effective enumeration of Turing machines, of a machine that halts on exactly the elements of $W$.

**Proposition 2.3** (folklore)**.** *The value of every rational probabilistic finite automaton is a left-c.e. real in* $[0, 1]$.

*Proof.* Enumerate $\Sigma^*$ as $w_0, w_1, \ldots$. The numbers $v_n = \max_{i \le n} P_A(w_i)$ are rationals in $[0, 1]$, computable from $n$ by exact arithmetic, nondecreasing, and they converge to $\sup_i P_A(w_i) = V(A)$. □

## 3 Counter programs that output words

**Definition 3.1.** A *counter program* $P$ with $d$ counters consists of a finite set $\mathcal{C}$ of control states with two distinct states $\mathsf{start}$ and $\mathsf{halt}$, and a finite set $\mathcal{I}$ of instructions. Each instruction $\iota$ has a source $\mathrm{src}(\iota) \in \mathcal{C} \setminus \{\mathsf{halt}\}$, a destination $\mathrm{dst}(\iota) \in \mathcal{C}$, a guard $g(\iota) \in \{0, +, *\}^d$, an increment $\delta(\iota) \in \{-1, 0, 1\}^d$ with $g_j(\iota) = +$ whenever $\delta_j(\iota) = -1$, and an output $o(\iota) \in \{0, 1, \varepsilon\}$. A *configuration* is a pair $(c, \mathbf{n}) \in \mathcal{C} \times \mathbb{N}^d$. The instruction $\iota$ is *enabled* at $(c, \mathbf{n})$ if $\mathrm{src}(\iota) = c$ and, for every $j$, $n_j = 0$ when $g_j(\iota) = 0$ and $n_j > 0$ when $g_j(\iota) = +$; it then leads to $(\mathrm{dst}(\iota), \mathbf{n} + \delta(\iota))$. A *run* from $\mathbf{m} \in \mathbb{N}^d$ is a sequence $(c_0, \mathbf{n}_0), \iota_1, (c_1, \mathbf{n}_1), \ldots, \iota_T, (c_T, \mathbf{n}_T)$ with $(c_0, \mathbf{n}_0) = (\mathsf{start}, \mathbf{m})$ and $c_T = \mathsf{halt}$, in which each $\iota_i$ is enabled at $(c_{i-1}, \mathbf{n}_{i-1})$ and leads to $(c_i, \mathbf{n}_i)$. Its output is the binary word $o(\iota_1) \cdots o(\iota_T)$, where $\varepsilon$ is the empty word. We write $L(P, \mathbf{m})$ for the set of outputs of runs from $\mathbf{m}$, and $L(P) = L(P, \mathbf{0})$.

The program may be nondeterministic: several instructions may be enabled at one configuration. We prove two lemmas. The first is Minsky's simulation of Turing machines by counter machines [22, 23], with a nondeterministic phase that writes the output; the second is its universal version.

**Lemma 3.2.** *For every computably enumerable set $W$ of binary words there is a counter program $P$ with three counters such that $L(P) = W$. The program can be computed from an enumeration index of $W$.*

**Lemma 3.3.** *There are a counter program $U$ with four counters $x_0, x_1, x_2, x_3$ and a sequence $(W_e)_{e\in\mathbb{N}}$ of sets of binary words such that $L(U, (e, 0, 0, 0)) = W_e$ for every $e$, and such that for every computably enumerable set $W$ of binary words some $e$, computable from an enumeration index of $W$, satisfies $W_e = W$.*

**Turing machines.** We use deterministic one-tape Turing machines $M$ with a finite set $K$ of states, a start state $k_0$, a halting state $k_h$, a finite tape alphabet $\Gamma$ containing 0, 1 and the blank $\Box$, and a transition function $\delta_M\colon (K \setminus \{k_h\}) \times \Gamma \to K \times \Gamma \times \{-1, 0, 1\}$. The cells of the tape are indexed by $\mathbb{Z}$. On input $y = y_0 \cdots y_{m-1}$ the machine starts in $k_0$ with $y_i$ in cell $i$, blanks elsewhere, and the head on cell 0; it *halts* on $y$ if it reaches $k_h$. A set of binary words is computably enumerable if and only if it is the set of binary words on which some such machine halts [29]. For a word $y$ we write $y^R$ for its reversal; the reversal of a computably enumerable set is computably enumerable.

**Stacks in counters.** Fix a machine $M$, let $h = |\Gamma| + 1$, and fix a bijection $\mathrm{code}\colon \Gamma \to \{1, \ldots, h-1\}$. A finite sequence $\gamma_1\gamma_2\cdots\gamma_r$ over $\Gamma$, read as a stack with top $\gamma_1$, is stored in a counter as

$$\ulcorner \gamma_1 \cdots \gamma_r \urcorner = \sum_{i=1}^{r} \mathrm{code}(\gamma_i)\, h^{i-1},$$

and the empty sequence is stored as 0. All base-$h$ digits of a stored number are nonzero, so the number determines the sequence. A configuration of $M$ in state $k$, with tape $\tau\colon \mathbb{Z} \to \Gamma$ and head on cell $z$, is *represented* by the control state $(k, \tau(z))$ together with counter values $x_1 = \ulcorner \tau(z-1)\cdots\tau(z-r)\urcorner$, $x_2 = \ulcorner \tau(z+1)\cdots\tau(z+r')\urcorner$ and $x_3 = 0$, for some $r, r' \geq 0$ such that every cell beyond those listed is blank. Different choices of $r, r'$ represent the same configuration.

**Lemma 3.4** (Stack operations)**.** *Let $i \in \{1, 2\}$ and $\gamma \in \Gamma$. There are finite sets of instructions without output, $\mathrm{PUSH}_i(\gamma)$ with an entry state and an exit state, and $\mathrm{POP}_i$ with an entry state and exit states $\mathrm{exit}_{\gamma'}$ for $\gamma' \in \Gamma$, with the following properties. At every control state of these sets other than the exits, exactly one instruction is enabled at each configuration. Started at the entry of $\mathrm{PUSH}_i(\gamma)$ with $x_3 = 0$ and $x_i$ storing a sequence $\sigma$, the instructions lead to the exit with $x_i$ storing $\gamma\sigma$, $x_3 = 0$ and the other counters unchanged. Started at the entry of $\mathrm{POP}_i$ with $x_3 = 0$ and $x_i$ storing $\sigma$, they lead to $\mathrm{exit}_{\gamma'}$, where $\gamma'$ is the top of $\sigma$, or $\Box$ if $\sigma$ is empty, with $x_i$ storing $\sigma$ without its top, $x_3 = 0$ and the other counters unchanged.*

*Proof.* For $\mathrm{PUSH}_i(\gamma)$ use control states $\eta, \eta_1, \ldots, \eta_{h-1}, \zeta, \zeta_1, \ldots, \zeta_{\mathrm{code}(\gamma)}$, with entry $\eta$. At $\eta$: if $x_i > 0$, decrement $x_i$, increment $x_3$ and go to $\eta_1$; if $x_i = 0$, go to $\zeta$. At $\eta_l$ for $l < h-1$: increment $x_3$ and go to $\eta_{l+1}$; at $\eta_{h-1}$: increment $x_3$ and go to $\eta$. At $\zeta$: if $x_3 > 0$, decrement $x_3$, increment $x_i$ and stay at $\zeta$; if $x_3 = 0$, go to $\zeta_1$. At $\zeta_l$ for $l < \mathrm{code}(\gamma)$: increment $x_i$ and go to $\zeta_{l+1}$; at $\zeta_{\mathrm{code}(\gamma)}$: increment $x_i$ and go to the exit. Each pass from $\eta$ back to $\eta$ moves one unit out of $x_i$ and $h$ units into $x_3$, so on reaching $\zeta$ we have $x_i = 0$ and $x_3 = h \cdot \ulcorner\sigma\urcorner$; the loop at $\zeta$ moves this back, and the chain $\zeta_1, \ldots, \zeta_{\mathrm{code}(\gamma)}$ adds $\mathrm{code}(\gamma)$. The result is $h\ulcorner\sigma\urcorner + \mathrm{code}(\gamma) = \ulcorner\gamma\sigma\urcorner$.

For $\mathrm{POP}_i$ use control states $\eta_0, \ldots, \eta_{h-1}$ and $\zeta_0, \ldots, \zeta_{h-1}$, with entry $\eta_0$. At $\eta_l$: if $x_i > 0$ and $l < h-1$, decrement $x_i$ and go to $\eta_{l+1}$; if $x_i > 0$ and $l = h-1$, decrement $x_i$, increment $x_3$ and go to $\eta_0$; if $x_i = 0$, go to $\zeta_l$. At $\zeta_l$: if $x_3 > 0$, decrement $x_3$, increment $x_i$ and stay at $\zeta_l$;

if $x_3 = 0$, go to $\text{exit}_{\gamma'}$ with $\text{code}(\gamma') = l$ if $l \geq 1$, and to $\text{exit}_\square$ if $l = 0$. By induction, after $X$ decrements of $x_i$ the program is at $\eta_{X \bmod h}$ with $x_3 = \lfloor X/h \rfloor$. Let $X_0 = \ulcorner \sigma \urcorner$. If $\sigma$ is empty, $X_0 = 0$ and the program goes from $\eta_0$ to $\zeta_0$ to $\text{exit}_\square$ with all counters unchanged. Otherwise $X_0 = h\ulcorner \sigma' \urcorner + \text{code}(\gamma')$, where $\gamma'$ is the top of $\sigma$ and $\sigma'$ the rest, so the decrements end at $\eta_l$ with $l = \text{code}(\gamma') \geq 1$ and $x_3 = \ulcorner \sigma' \urcorner$, and the loop at $\zeta_l$ moves $\ulcorner \sigma' \urcorner$ back into $x_i$. In both sets the enabled instruction at each state is unique, because the alternatives have complementary guards. □

*Proof of Lemma 3.2.* Let $M$ be a Turing machine that halts on exactly the binary words in $W^R = \{s^R : s \in W\}$. The program $P$ has the counters $x_1, x_2, x_3$, the control states start, halt, the pairs $(k, \gamma) \in K \times \Gamma$, and the states of the copies of the stack operations used below, each copy with its own states. Its instructions are:

(G1) for $v \in \{0, 1\}$, an instruction from start with output $v$ to the entry of a copy of $\text{PUSH}_2(v)$ whose exit is start;

(G2) an instruction from start to the entry of a copy of $\text{POP}_2$ whose exit $\text{exit}_\gamma$ is $(k_0, \gamma)$ for each $\gamma$;

(S) for $k \neq k_h$ and $\gamma \in \Gamma$ with $\delta_M(k, \gamma) = (k', \gamma', D)$: if $D = 0$, an instruction from $(k, \gamma)$ to $(k', \gamma')$; if $D = +1$, an instruction from $(k, \gamma)$ to a copy of $\text{PUSH}_1(\gamma')$ whose exit is the entry of a copy of $\text{POP}_2$ with exits $(k', \cdot)$; if $D = -1$, the same with $\text{PUSH}_2(\gamma')$ and $\text{POP}_1$;

(H) for each $\gamma$, an instruction from $(k_h, \gamma)$ to halt.

Only the instructions (G1) have an output. Instructions outside the copies of the stack operations have guard $(*, *, *)$ and increment 0.

Consider a run from $\mathbf{0}$. At start the run uses (G1) some number $m \geq 0$ of times, with bits $s_1, \ldots, s_m$, each followed by its copy of $\text{PUSH}_2$, and then (G2). By Lemma 3.4, before (G2) we have $x_1 = x_3 = 0$ and $x_2 = \ulcorner s_m s_{m-1} \cdots s_1 \urcorner$, and after the copy of $\text{POP}_2$ the control state is $(k_0, \gamma)$ with $\gamma = s_m$, or $\gamma = \square$ if $m = 0$, and $x_2 = \ulcorner s_{m-1} \cdots s_1 \urcorner$. This represents the start of $M$ on the input $s_m \cdots s_1 = s^R$, where $s = s_1 \cdots s_m$. From then on exactly one instruction is enabled at each configuration, and by Lemma 3.4 a pass through one instruction (S) and its copies maps a representation of a configuration of $M$ to a representation of the next configuration: for $D = +1$, pushing $\gamma'$ onto $x_1$ lists the cells to the left of the new head position, and popping $x_2$ yields the symbol under it, or a blank. Hence the run reaches halt if and only if $M$ halts on $s^R$, that is, if and only if $s \in W$. Its output is $s$, because only (G1) has an output. Conversely, for each $s \in W$ the choices above give a run with output $s$. So $L(P) = W$. Every step is effective in $M$, and $M$ is computable from an enumeration index of $W$. □

*Proof of Lemma 3.3.* Fix an effective enumeration $(M_e)_{e \in \mathbb{N}}$ of the Turing machines above. By the existence of a universal Turing machine, which goes back to Turing [31] (for the formulation with halting used here see, for example, [23, 29]), there is one Turing machine $T_U$, whose tape alphabet contains a symbol $\sharp$, such that for all $e \in \mathbb{N}$ and all binary words $y$, $T_U$ halts on the input $1^e \sharp y$ if and only if $M_e$ halts on $y$. Let $W_e = \{s : M_e \text{ halts on } s^R\}$. If $W$ is computably enumerable, so is $W^R$, and any $e$ such that $M_e$ halts on exactly the binary words in $W^R$ satisfies $W_e = W$; such an $e$ is computable from an index of $W$.

The program $U$ has the counters $x_0, \ldots, x_3$, where $x_0$ is only used as described now, and the instructions (G1), (S) and (H) of the proof of Lemma 3.2, with $T_U$ in place of $M$. Instead of (G2) it has an instruction from start to a copy of $\text{PUSH}_2(\sharp)$ whose exit is a new control state init, and at init: if $x_0 > 0$, an instruction that decrements $x_0$ and goes to a copy of $\text{PUSH}_2(1)$

whose exit is init; if $x_0 = 0$, an instruction to a copy of $\mathrm{POP}_2$ with exits $(k_0, \cdot)$. A run from $(e, 0, 0, 0)$ with output $s = s_1 \cdots s_m$ reaches init with $x_2 = \ulcorner \sharp s_m \cdots s_1 \urcorner$, leaves it with $x_0 = 0$ and $x_2 = \ulcorner 1^e \sharp s_m \cdots s_1 \urcorner$, and after the copy of $\mathrm{POP}_2$ represents the start of $T_U$ on $1^e \sharp s^R$. As before, the run reaches halt if and only if $T_U$ halts on $1^e \sharp s^R$, that is, if and only if $s \in W_e$. The instructions of $U$ do not depend on $e$. □

# 4 Checking a run with two random tests

## 4.1 The equality test

**Lemma 4.1.** *Let $B \geq 2$ be an integer and $\kappa_B = (B + B^{-1})/2$. For $n, n' \in \mathbb{N}$ put*

$$\varphi_B(n, n') = B^{-(n+n')}, \qquad \psi_B(n, n') = \tfrac{1}{2}\big(B^{-2n} + B^{-2n'}\big).$$

*Then $\psi_B(n, n') = \varphi_B(n, n')$ if $n = n'$, and $\psi_B(n, n') \geq \kappa_B\, \varphi_B(n, n')$ if $n \neq n'$.*

*Proof.* The ratio is $\psi_B / \varphi_B = g(n - n')$ with $g(t) = (B^t + B^{-t})/2$. The function $g$ is even, $g(0) = 1$, and for integers $t \geq 0$

$$g(t+1) - g(t) = \tfrac{1}{2}(B-1)\big(B^t - B^{-t-1}\big) > 0.$$

Hence $g(t) \geq g(1) = \kappa_B$ for every integer $t \neq 0$. □

A finite automaton realizes $\varphi_B$ by surviving each unary symbol of both lengths with probability $1/B$, and $\psi_B$ by choosing one of the two lengths with a fair coin and surviving each of its symbols with probability $1/B^2$. Since $\kappa_B \to \infty$, a single false equation makes the second experiment much more likely to survive than the first.

## 4.2 Certificates

Fix a counter program $P$ with $d$ counters. Its certificate alphabet is

$$\Sigma_P = \{[\iota] : \iota \in \mathcal{I}\} \cup \{\dashv, \mathtt{u}, \diamond, \#\}.$$

For $\mathbf{n} \in \mathbb{N}^d$ let $\langle \mathbf{n} \rangle = \mathtt{u}^{n_1} \diamond \mathtt{u}^{n_2} \diamond \cdots \mathtt{u}^{n_d} \diamond$, the $d$ *blocks* of $\mathbf{n}$.

**Definition 4.2.** A *certificate* is a word

$$t = [\iota_1]\langle \mathbf{n}_0 \rangle [\iota_2] \langle \mathbf{n}_1 \rangle \cdots [\iota_T] \langle \mathbf{n}_{T-1} \rangle \dashv \langle \mathbf{n}_T \rangle \#$$

with $T \geq 0$, $\iota_i \in \mathcal{I}$ and $\mathbf{n}_i \in \mathbb{N}^d$. Its *rows* are the words $[\iota_{i+1}]\langle \mathbf{n}_i \rangle$ for $i < T$ and $\dashv \langle \mathbf{n}_T \rangle$; row $i$ has the *head* $[\iota_{i+1}]$ or $\dashv$. Its *output* is $\mathrm{out}(t) = o(\iota_1) \cdots o(\iota_T)$. It is *well formed* if

(W1) $T \geq 1$, $\mathrm{src}(\iota_1) = \mathsf{start}$, $\mathrm{src}(\iota_{i+1}) = \mathrm{dst}(\iota_i)$ for $1 \leq i < T$, and $\mathrm{dst}(\iota_T) = \mathsf{halt}$;

(W2) $\mathbf{n}_0 = \mathbf{0}$;

(W3) for $0 \leq i < T$, the vector $\mathbf{n}_i$ satisfies the guard of $\iota_{i+1}$.

Its *equations* are the pairs, for $0 \leq i < T$ and $1 \leq j \leq d$,

$$E_{i,j} = \big(n_{i,j} + \delta^+,\ n_{i+1,j} + \delta^-\big), \qquad \delta = \delta_j(\iota_{i+1}),$$

where $\delta^+ = \max(\delta, 0)$ and $\delta^- = \max(-\delta, 0)$; the two numbers are its *members*. A well-formed certificate is *valid* if the two members of every equation are equal.

The members of $E_{i,j}$ are equal if and only if $n_{i+1,j} = n_{i,j} + \delta_j(\iota_{i+1})$. To a run $\rho$ from $\mathbf{0}$ as in Definition 3.1 we associate the certificate $\mathrm{cert}(\rho) = [\iota_1]\langle \mathbf{n}_0 \rangle \cdots [\iota_T]\langle \mathbf{n}_{T-1} \rangle \dashv \langle \mathbf{n}_T \rangle \#$.

**Lemma 4.3.** *A certificate is valid if and only if it is* $\mathrm{cert}(\rho)$ *for a run* $\rho$ *of* $P$ *from* $\mathbf{0}$*, and then its output is the output of* $\rho$*.*

*Proof.* Put $c_0 = \mathsf{start}$ and $c_i = \mathrm{dst}(\iota_i)$. Conditions (W1)–(W3) and the equalities $\mathbf{n}_{i+1} = \mathbf{n}_i + \delta(\iota_{i+1})$ say exactly that $(c_0, \mathbf{n}_0), \iota_1, \ldots, \iota_T, (c_T, \mathbf{n}_T)$ is a run from $\mathbf{0}$. The outputs agree by definition. □

### 4.3 The parser

**Lemma 4.4.** *There is a deterministic finite automaton over* $\Sigma_P$*, the* parser*, with an initial state* $p_\circ$ *and an absorbing error state* $p_{\mathrm{err}}$*, such that for every word* $w$ *the parser started in* $p_\circ$ *is in* $p_\circ$ *after reading* $w$ *if and only if* $w$ *is a concatenation of well-formed certificates, and it does not enter* $p_{\mathrm{err}}$ *while it reads a prefix of such a concatenation. While it reads a letter* $\mathtt{u}$ *or* $\diamond$*, its state determines the head of the current row, the index* $j$ *of the current block, and whether the current row is row* 0*.*

*Proof.* A state of the parser outside $\{p_\circ, p_{\mathrm{err}}\}$ records the control state $c$ that the next head must have as its source, the head of the current row, the index $j \in \{1, \ldots, d+1\}$ of the current block ($d+1$ once the row is complete), whether the current block has a symbol $\mathtt{u}$ yet, and whether the current row is row 0. In $p_\circ$ the expected source is start, no row has started, and the next row is row 0. On a head $[\iota]$ at the start of a row, the parser checks $\mathrm{src}(\iota) = c$, records $\iota$ and sets $c = \mathrm{dst}(\iota)$. On $\dashv$ at the start of a row it checks $c = \mathsf{halt}$, which excludes $T = 0$ because $\mathsf{start} \neq \mathsf{halt}$, and then replaces $c$ by a mark that is the source of no instruction and is not halt; so after a row with head $\dashv$ is complete, the only allowed letter is #. On $\mathtt{u}$ it checks that the current row is not row 0 and that the guard of the head does not require the current counter to be zero; for these checks the head $\dashv$ of a final row counts as having a guard with no requirements. On $\diamond$ it checks that the guard does not require the current counter to be positive when the block is empty, and moves to the next block. On # it checks that the current row is complete and has head $\dashv$, and returns to $p_\circ$. Every other letter, for instance $\mathtt{u}$ or $\diamond$ after the last block of a row, and every failed check, leads to $p_{\mathrm{err}}$. These checks are exactly (W1)–(W3) and the format of Definition 4.2, and each is made at the letter that completes the condition checked, so a prefix of a concatenation of well-formed certificates passes all of them. □

### 4.4 The checking automaton

Fix $P$ and an integer $B \geq 2$. The checking automaton has the following states. A *live* state is either $(p, \mathsf{j}, r)$ or $(p, \mathsf{s}, \xi, r)$, where $p \neq p_{\mathrm{err}}$ is a parser state, j and s mark the joint and the split test, $r \in \{0, 1\}$ is the *register*, and $\xi = (\xi_j^{\mathrm{in}}, \xi_j^{\mathrm{out}})_{j \leq d}$ with entries in $\{\emptyset, 1, 2\}$ records the choices of the split test. A *dead* state is $(p, \dagger)$ with $p \neq p_{\mathrm{err}}$. There are two further states, yes and no. The initial distribution $\pi_\circ$ gives probability $1/2$ to $(p_\circ, \mathsf{j}, 0)$ and $1/2$ to $(p_\circ, \mathsf{s}, \emptyset, 0)$, where $\emptyset$ denotes the vector with all entries $\emptyset$.

In a block $j$ of row $i$, let $\chi_j \in \{0, 1, 2\}$ be the number of equations of which this block is a member: $\chi_j = [i \geq 1] + [\text{the head of row } i \text{ is not } \dashv]$. By Lemma 4.4 the parser state determines $\chi_j$. For a letter $\ell \neq \#$, let $p'$ be the parser state after $\ell$, and let $o(\ell) = o(\iota)$ if $\ell = [\iota]$ and $o(\ell) = \varepsilon$ otherwise. The two-state register matrices are

$$R_0 = \begin{pmatrix} 1 & 0 \\ 1/2 & 1/2 \end{pmatrix}, \qquad R_1 = \begin{pmatrix} 1/2 & 1/2 \\ 0 & 1 \end{pmatrix}, \qquad R_\varepsilon = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

in the state order $0, 1$. On $\ell \neq \#$ the checking automaton acts as follows.

(C1) If $p' = p_{\mathrm{err}}$, every live or dead state moves to no.

(C2) Otherwise a dead state $(p, \dagger)$ moves to $(p', \dagger)$. A live state moves its parser state to $p'$ and its register from $r$ to $r'$ with probability $R_{o(\ell)}(r, r')$, independently of everything else, and it survives with the probability given below; if it does not survive it moves to $(p', \dagger)$.

(C3) Joint test. On $[\iota]$ it survives with probability $B^{-\sum_j |\delta_j(\iota)|}$; on u in block $j$ with probability $B^{-\chi_j}$; on other letters with probability 1.

(C4) Split test. On $[\iota]$ it draws $\xi_j^{\mathrm{out}} \in \{1, 2\}$ uniformly and independently for $j = 1, \dots, d$, and survives with probability

$$\prod_{j=1}^{d} B^{-2\left(\delta_j^+ [\xi_j^{\mathrm{out}}=1] + \delta_j^- [\xi_j^{\mathrm{out}}=2]\right)}, \qquad \delta_j = \delta_j(\iota).$$

On u in block $j$ it survives with probability $B^{-2([\xi_j^{\mathrm{in}}=2]+[\xi_j^{\mathrm{out}}=1])}$. On the separator $\diamond$ that closes block $j$ it survives with probability 1 and replaces $(\xi_j^{\mathrm{in}}, \xi_j^{\mathrm{out}})$ by $(\xi_j^{\mathrm{out}}, \varnothing)$. On other letters it survives with probability 1 and keeps $\xi$.

Here $[\cdot]$ is 1 if the condition holds and 0 otherwise. The choice $\xi_j^{\mathrm{out}}$ drawn at the head of row $i$ is the choice of the split test for the equation $E_{i,j}$: the value 1 selects its first member and 2 its second. It is kept in $\xi_j^{\mathrm{out}}$ while block $j$ of row $i$ is read, then in $\xi_j^{\mathrm{in}}$ while block $j$ of row $i+1$ is read, and then discarded. At each moment at most two equations per counter have a recorded choice, so the state set is finite. All probabilities are rational, and each row sums to 1. The rows of yes and no, and the action of #, are given in Section 5.

For a well-formed certificate $t = t'\#$ with equations $\mathcal{E}(t) = \{E_{i,j}\}$, put

$$\Phi(t) = \frac{1}{2} \prod_{(n,n') \in \mathcal{E}(t)} \varphi_B(n, n'), \qquad \Psi(t) = \frac{1}{2} \prod_{(n,n') \in \mathcal{E}(t)} \psi_B(n, n').$$

For a binary word $s = s_1 \cdots s_k$ let $x(s) = \sum_{i=1}^{k} s_i 2^{i-k-1}$, so that $x(\varepsilon) = 0$ and $x(sv) = (x(s) + v)/2$ for a bit $v$.

**Lemma 4.5.** *Let $t = t'\#$ be a well-formed certificate with output $s$. Started from $\pi_\circ$, after reading $t'$ the checking automaton is in live joint states with total probability $\Phi(t)$, in live split states with total probability $\Psi(t)$, and in dead states otherwise. Within each of the two live parts, the register is 1 with conditional probability $x(s)$.*

*Proof.* No letter of $t'$ leads the parser to $p_{\mathrm{err}}$, so (C1) does not occur, and the parser state is the same for all states at every moment.

Joint test. Its survival probability is the product of the factors in (C3). A symbol u in block $j$ of row $i$ contributes $B^{-1}$ for each equation of which the block is a member, and the head $[\iota_{i+1}]$ contributes $B^{-1}$ for each unit of the offsets $\delta^+$ and $\delta^-$ in the equations $E_{i,j}$, since $\delta^+ + \delta^- = |\delta|$. The product is therefore $\prod_{(n,n') \in \mathcal{E}(t)} B^{-(n+n')}$, and with the initial factor $1/2$ the probability is $\Phi(t)$.

Split test. Fix the choices $\xi_{i,j} \in \{1, 2\}$ for all equations. Given these, the survival probability is the product of the factors in (C4). By the description of the slots above, a symbol u in block $j$ of row $i$ contributes $B^{-2}$ if $\xi_{i-1,j} = 2$ (it lies in the second member of $E_{i-1,j}$, which was chosen) and $B^{-2}$ if $\xi_{i,j} = 1$ (it lies in the first member of $E_{i,j}$, which was chosen); the head of

row $i$ contributes $B^{-2\delta^+}$ if $\xi_{i,j} = 1$ and $B^{-2\delta^-}$ if $\xi_{i,j} = 2$, for each $j$. Collecting the factors by equation, the survival probability given the choices is $\prod_{i,j} f_{i,j}(\xi_{i,j})$, where $f_{i,j}(1) = B^{-2m}$ and $f_{i,j}(2) = B^{-2m'}$ for $E_{i,j} = (m, m')$. The choices are independent and uniform, so averaging gives $\prod_{i,j} \frac{1}{2}(f_{i,j}(1) + f_{i,j}(2)) = \prod_{(m,m')\in\mathcal{E}(t)} \psi_B(m, m')$; with the initial factor $1/2$ the probability is $\Psi(t)$.

Register. The register moves by $R_{o(\ell)}$ independently of the other components, and its matrices depend only on the letter, so the joint law of the register and the other components remains a product law. If the register is 1 with probability $x$, then after $R_0$ it is 1 with probability $x/2$ and after $R_1$ with probability $(1-x)/2 + x = (1+x)/2$. Starting from 0, after the heads of $t'$ it is 1 with probability $x(s)$. □

**Corollary 4.6.** *Let $t$ be a well-formed certificate. Then $\Phi(t) > 0$. If $t$ is valid, $\Psi(t) = \Phi(t)$. If $t$ is not valid, $\Psi(t) \geq \kappa_B \Phi(t)$.*

*Proof.* All factors are positive. By Lemma 4.1, every factor ratio $\psi_B/\varphi_B$ is at least 1, equal to 1 for an equation with equal members, and at least $\kappa_B$ for an equation with unequal members. A certificate that is well formed but not valid has at least one such equation. □

## 5 The payout and the proof of Theorem A

Fix rationals $0 < a < b < 1$ and put $\beta_0 = a$ and $\beta_1 = b$. For a binary word $s$ let $q(s) = a + (b-a)x(s)$, so $q(s) \in [a, b)$. The automaton $A(P, a, b, B)$ consists of the checking automaton, with initial distribution $\pi_\circ$ and accepting set $F = \{\mathsf{yes}\}$, and the following transitions.

(P1) The states yes and no are absorbing under every letter.

(P2) On #, if the parser moves to $p_{\mathrm{err}}$, every live or dead state moves to no.

(P3) On #, if the parser moves to $p_\circ$, that is, at the end of a well-formed certificate: a live joint state with register $r$ moves to yes with probability $\beta_r$; a live split state with register $r$ moves to no with probability $1 - \beta_r$; all remaining probability of a live state, and all probability of a dead state, is distributed according to $\pi_\circ$.

The rows in (P3) are rational combinations of a point mass and $\pi_\circ$, so $A(P, a, b, B)$ is a rational probabilistic finite automaton. The only accepting state is yes, which is entered only in (P3).

**Lemma 5.1.** *Let $t$ be a well-formed certificate with output $s$. Started from $\pi_\circ$, after reading $t$ the automaton is in* yes *with probability* $\mathrm{acc}(t) = \Phi(t)q(s)$, *in* no *with probability* $\mathrm{rej}(t) = \Psi(t)(1 - q(s))$, *and the remaining probability* $1 - \mathrm{acc}(t) - \mathrm{rej}(t)$ *is distributed according to $\pi_\circ$.*

*Proof.* By Lemma 4.5, before the final # the live joint states carry probability $\Phi(t)$, with the register 1 with conditional probability $x(s)$. By (P3) they move to yes with probability $\Phi(t)\big((1-x(s))a + x(s)b\big) = \Phi(t)q(s)$. In the same way the live split states move to no with probability $\Psi(t)\big((1-x(s))(1-a) + x(s)(1-b)\big) = \Psi(t)(1-q(s))$. Everything else is redistributed according to $\pi_\circ$. □

**Lemma 5.2.** *Let $\theta \in [0, 1]$, and suppose that $\mathrm{acc}(t) \leq \theta\,(\mathrm{acc}(t) + \mathrm{rej}(t))$ for every well-formed certificate $t$. Then $P_A(w) \leq \theta$ for every word $w \in \Sigma_P^*$.*

*Proof.* A well-formed certificate contains exactly one #, at its end, so every word can be written uniquely as $w = t_1 \cdots t_m z$, where $t_1, \ldots, t_m$ are well-formed certificates and no prefix of $z$ is a well-formed certificate. Put $\mu_1 = 1$ and $\mu_{i+1} = \mu_i(1 - \mathrm{acc}(t_i) - \mathrm{rej}(t_i))$. By Lemma 5.1 and

linearity, after reading $t_1 \cdots t_i$ the automaton is in yes with probability $\sum_{l \le i} \mu_l \operatorname{acc}(t_l)$, and the probability outside $\{\mathsf{yes}, \mathsf{no}\}$ equals $\mu_{i+1}$ and is distributed according to $\pi_\circ$. While $z$ is read, the parser starts in $p_\circ$, and a transition into yes would require it to complete a well-formed certificate, which does not happen; the probability in yes does not decrease, by (P1). Hence

$$P_A(w) = \sum_{i=1}^{m} \mu_i \operatorname{acc}(t_i) \le \theta \sum_{i=1}^{m} \mu_i \big(\operatorname{acc}(t_i) + \operatorname{rej}(t_i)\big) = \theta \sum_{i=1}^{m} (\mu_i - \mu_{i+1}) = \theta(1 - \mu_{m+1}) \le \theta. \quad \square$$

**Lemma 5.3.** *Let $t$ be a valid certificate with output $s$. Then $P_A(t^m) = q(s)\big(1 - (1 - \Phi(t))^m\big)$ for every $m \ge 1$.*

*Proof.* By Corollary 4.6, $\Psi(t) = \Phi(t)$, so $\operatorname{acc}(t) + \operatorname{rej}(t) = \Phi(t)$. By Lemma 5.1, as in the proof of Lemma 5.2, $P_A(t^m) = \sum_{i=1}^{m} (1 - \Phi(t))^{i-1} \Phi(t) q(s) = q(s)\big(1 - (1 - \Phi(t))^m\big)$. $\square$

**Theorem 5.4.** *Let $\alpha \in (0,1)$ be left-c.e., let $a, b$ be rationals with $0 < a < \alpha < b < 1$, and let $B \ge 2$ be an integer with*

$$\kappa_B > \frac{b(1-a)}{a(1-b)}. \tag{1}$$

*Let $P$ be a counter program with $L(P) = W_\alpha$, where $W_\alpha = \{s \in \{0,1\}^* : q(s) < \alpha\}$. Then $V(A(P, a, b, B)) = \alpha$. Consequently Theorem A holds.*

*Proof.* The set $W_\alpha$ is computably enumerable: given $s$, compute the rational $q(s)$ and search for $n$ with $\alpha_n > q(s)$, where $(\alpha_n)$ is a computable nondecreasing sequence of rationals converging to $\alpha$. So $P$ exists by Lemma 3.2, and $B$ exists because $\kappa_B \to \infty$. Write $A = A(P, a, b, B)$.

Upper bound. Let $t$ be a well-formed certificate with output $s$. By Corollary 4.6, $\Phi(t) > 0$ and $\operatorname{acc}(t) + \operatorname{rej}(t) \ge \Phi(t) q(s) > 0$. If $t$ is valid, then $s \in L(P) = W_\alpha$ by Lemma 4.3, and $\operatorname{acc}(t)/(\operatorname{acc}(t) + \operatorname{rej}(t)) = q(s) < \alpha$. If $t$ is not valid, then $\Psi(t) \ge \kappa_B \Phi(t)$, and since $f(q) = q/(q + \kappa_B(1-q))$ is increasing on $[0,1]$ and $q(s) \le b$,

$$\frac{\operatorname{acc}(t)}{\operatorname{acc}(t) + \operatorname{rej}(t)} \le \frac{q(s)}{q(s) + \kappa_B(1 - q(s))} \le \frac{b}{b + \kappa_B(1-b)} < a < \alpha;$$

the strict inequality is a rearrangement of (1). By Lemma 5.2 with $\theta = \alpha$, $P_A(w) \le \alpha$ for every word $w$.

Lower bound. Let $s \in W_\alpha$. By Lemma 4.3 some valid certificate $t$ has output $s$, and by Lemma 5.3 $V(A) \ge \sup_m P_A(t^m) = q(s)$. As $s$ ranges over binary words, $x(s)$ ranges over all dyadic rationals in $[0,1)$. The number $\vartheta = (\alpha - a)/(b - a)$ lies in $(0,1)$, and $q(s) < \alpha$ if and only if $x(s) < \vartheta$. Dyadic rationals below $\vartheta$ come arbitrarily close to $\vartheta$, so $\sup_{s \in W_\alpha} q(s) = a + (b-a)\vartheta = \alpha$, and $V(A) \ge \alpha$.

Theorem A. Every value is left-c.e. in $[0,1]$ by Proposition 2.3. Every left-c.e. $\alpha \in (0,1)$ has rationals $a, b$ as above and is a value by the first part. The values 0 and 1 are those of the one-state automata with $F = \emptyset$ and $F = Q$. $\square$

## 6 An explicit noncomputable value and halting probabilities

Let $(M_n)$ be the fixed effective enumeration of Turing machines of Lemma 3.3, let $H$ be the set of $n$ such that $M_n$ halts on input $n$, written in binary, and put $\omega_n = 4^{-(n+2)}$ and $\alpha_* = 1/4 + \sum_{n \in H} \omega_n$. The set $H$ is computably enumerable and, by the diagonal argument, not computable (see [29]).

**Lemma 6.1.** *Let $\alpha_t = 1/4 + \sum\{\omega_n : n \le t,\ M_n \text{ halts on input } n \text{ within } t \text{ steps}\}$. The sequence $(\alpha_t)$ is computable, nondecreasing and converges to $\alpha_*$, and $1/4 \le \alpha_* \le 1/3$. The real $\alpha_*$ is not computable.*

*Proof.* The first assertions are clear, and $\sum_{n\ge 0} \omega_n = (1/16)/(1-1/4) = 1/12$. For every $n$, $\sum_{l>n} \omega_l = \omega_n/3$. Suppose that rational approximations to $\alpha_*$ with any prescribed error were computable. We decide whether $n \in H$ by induction on $n$. Given $H \cap \{0,\dots,n-1\}$, the residual $\alpha_* - 1/4 - \sum_{l<n,\, l\in H} \omega_l$ is at most $\omega_n/3$ if $n \notin H$ and at least $\omega_n$ if $n \in H$. Subtract the same rational from an approximation of $\alpha_*$ with error less than $\omega_n/12$: the result is at most $5\omega_n/12$ in the first case and at least $11\omega_n/12$ in the second, and comparing it with $\omega_n/2$ decides whether $n \in H$. This would make $H$ computable. □

Let $T_*$ be the Turing machine that, on a binary input $y$, computes the rational $q = 1/8 + (3/8)\cdot 0.y$, where $0.y$ is the binary fraction with the digits of $y$ in order, and then computes $\alpha_0, \alpha_1, \dots$ and halts as soon as $\alpha_t > q$. Since $x(s) = 0.s^R$ for every binary word $s$, the machine $T_*$ halts on $s^R$ if and only if $1/8 + (3/8)x(s) < \alpha_*$. Let $P_*$ be the program of Lemma 3.2 built from $M = T_*$, so that $L(P_*) = \{s : 1/8 + (3/8)x(s) < \alpha_*\}$, and let $A_* = A(P_*, 1/8, 1/2, 16)$.

**Theorem 6.2.** *$V(A_*) = \alpha_*$, and $\alpha_*$ is not computable. In particular Theorem C holds.*

*Proof.* We check the hypotheses of Theorem 5.4 with $a = 1/8$, $b = 1/2$ and $B = 16$. By Lemma 6.1, $\alpha_*$ is left-c.e. and $a < 1/4 \le \alpha_* \le 1/3 < b$. Here $b(1-a)/(a(1-b)) = 7$ and $\kappa_{16} = 257/32 > 7$; correspondingly $b/(b + \kappa_{16}(1-b)) = 32/289 < 1/8$. With these constants $q(s) = 1/8 + (3/8)x(s)$ and $L(P_*) = W_{\alpha_*}$. Theorem 5.4 gives $V(A_*) = \alpha_*$, and Lemma 6.1 shows that $\alpha_*$ is not computable. □

For each fixed real $\alpha$, $\alpha$ is computable if and only if its lower cut $\{\lambda \in \mathbb{Q} : \lambda < \alpha\}$ is decidable; the equivalence is not uniform in an index of $\alpha$. Since $V(A_*) > \lambda$ if and only if $P_{A_*}(w) > \lambda$ for some word $w$, the strict emptiness problem for the fixed automaton $A_*$, with the rational threshold $\lambda$ as its only input, is undecidable.

A *prefix-free machine* is a partial computable function $\mathcal{M}$ from binary words to binary words whose domain is prefix free; its halting probability is $\Omega_{\mathcal{M}} = \sum_{\sigma \in \operatorname{dom}\mathcal{M}} 2^{-|\sigma|}$.

**Corollary 6.3.** *For every prefix-free machine $\mathcal{M}$ there is a rational probabilistic finite automaton with two input letters and at most $N$ states whose value is $\Omega_{\mathcal{M}}$, where $N$ is the number of Theorem B. In particular Corollary D holds.*

*Proof.* The words $\sigma$ in the domain correspond to the intervals $[0.\sigma, 0.\sigma + 2^{-|\sigma|})$ of length $2^{-|\sigma|}$, which are pairwise disjoint because no word of the domain is a prefix of another. Hence every finite partial sum of $\Omega_{\mathcal{M}}$ is at most 1. Enumerating the domain and adding $2^{-|\sigma|}$ for each new word gives a computable nondecreasing sequence of rationals converging to $\Omega_{\mathcal{M}}$, so $\Omega_{\mathcal{M}}$ is a left-c.e. real in $[0,1]$, and Theorem B applies. For a universal prefix-free machine, $\Omega_{\mathcal{M}}$ is Chaitin's halting probability [7]. □

*Remark* 6.4. By results of Chaitin [7], of Calude, Hertling, Khoussainov and Wang [6], and of Kučera and Slaman [18] (see [10]), the left-c.e. reals that are Martin-Löf random are exactly the halting probabilities of universal prefix-free machines. With Theorem A, the Martin-Löf random values of rational probabilistic finite automata are exactly these numbers.

*Remark* 6.5. Conversely, it is well known that every left-c.e. $\alpha \in [0,1]$ is $\Omega_{\mathcal{M}}$ for some prefix-free machine $\mathcal{M}$ [6, 10]; we recall the argument. Let $(\alpha_n)$ be a computable nondecreasing sequence of rationals with limit $\alpha$; replacing $\alpha_n$ by $\max(0, \alpha_{n-1})$ for $n \ge 1$ and setting $\alpha_0 = 0$,

we may assume $\alpha_0 = 0$ and $\alpha_n \geq 0$. The dyadic rationals $\lfloor 2^n \alpha_n \rfloor / 2^n$ also form a nondecreasing sequence with limit $\alpha$; their increments are finite sums of powers $2^{-k}$ with total $\alpha \leq 1$, and the Kraft–Chaitin theorem [7, 10] gives a prefix-free machine whose domain has one word of length $k$ for each such power. So Theorem A says that the values of rational probabilistic finite automata are exactly the halting probabilities of prefix-free machines.

## 7 Two letters and a fixed number of states

**Lemma 7.1.** *Let $A$ be a rational probabilistic finite automaton with $n$ states and $m \geq 1$ letters, and let $l \geq 1$ satisfy $2^l \geq m$. There is a rational probabilistic finite automaton $A'$ over $\{0,1\}$ with $n(2^l - 1) + 1$ states and $V(A') = V(A)$.*

*Proof.* Fix distinct codes $\mathrm{code}(\ell) \in \{0,1\}^l$ for the letters of $A$. The states of $A'$ are the pairs $(p, \sigma)$ of a state $p$ of $A$ and a binary word $\sigma$ of length less than $l$, and a rejecting sink. The initial distribution gives $(p, \varepsilon)$ the probability $\pi(p)$, and $(p, \sigma)$ is accepting if and only if $p \in F$. On a bit $y$, the state $(p, \sigma)$ moves to $(p, \sigma y)$ if $|\sigma y| < l$; to $(p', \varepsilon)$ with probability $M_\ell(p, p')$ if $\sigma y = \mathrm{code}(\ell)$; and to the sink if $\sigma y$ has length $l$ and is not a code. The sink is absorbing. A binary word that contains no invalid code is a concatenation $\mathrm{code}(\ell_1) \cdots \mathrm{code}(\ell_k)\sigma$ with $|\sigma| < l$, and it is accepted by $A'$ with probability $P_A(\ell_1 \cdots \ell_k)$, because the accepting set is read from the first component in every phase. A word that contains an invalid code is accepted with probability 0. Hence $V(A') \leq V(A)$, and $P_{A'}(\mathrm{code}(\ell_1) \cdots \mathrm{code}(\ell_k)) = P_A(\ell_1 \cdots \ell_k)$ gives $V(A') \geq V(A)$. □

**The index check.** Let $U$ be the program of Lemma 3.3, with $d = 4$ counters indexed $0, \ldots, 3$, and fix $e \in \mathbb{N}$. We modify the construction of Sections 4 and 5 so that it checks runs of $U$ from $(e, 0, 0, 0)$ without storing $e$ in the state.

(I1) Condition (W2) becomes $n_{0,1} = n_{0,2} = n_{0,3} = 0$, with $n_{0,0}$ arbitrary; the parser allows symbols $\mathtt{u}$ in block 0 of row 0.

(I2) Every certificate has one more equation $E_* = (e, n_{0,0})$, whose first member is the number $e$ itself.

(I3) In block 0 of row 0, $\chi_0 = 1 + [\text{the head of row 0 is not } \dashv]$. The choice of the split test for $E_*$ is kept in $\xi_0^{\mathrm{in}}$ during row 0.

(I4) A new initial distribution $\pi_\circ^e$ replaces $\pi_\circ$ everywhere, including in (P3). It gives the probability $B^{-e}/2$ to $(p_\circ, \mathsf{j}, 0)$, the probability $1/4$ to the state $(p_\circ, \mathsf{s}, \xi, 0)$ with $\xi_0^{\mathrm{in}} = 2$ and all other entries $\emptyset$, the probability $B^{-2e}/4$ to the same state with $\xi_0^{\mathrm{in}} = 1$, and the remaining probability, $(1 - B^{-e})/2 + (1 - B^{-2e})/4$, to the dead state $(p_\circ, \dagger)$.

In the joint test, (I4) contributes the factor $B^{-e}$ and (I3) the factor $B^{-1}$ for each symbol of block 0 of row 0 on account of $E_*$, so $E_*$ contributes $\varphi_B(e, n_{0,0})$. In the split test, choice 1 contributes $B^{-2e}$ through (I4) and nothing in block 0, and choice 2 contributes $B^{-2}$ for each symbol of that block, so $E_*$ contributes $\psi_B(e, n_{0,0})$. Write $A_U(e, a, b, B)$ for the resulting automaton.

**Lemma 7.2.** *With these modifications, Lemma 4.3 holds with runs from $(e, 0, 0, 0)$ in place of runs from $\mathbf{0}$, where valid means that both members of every equation, $E_*$ included, are equal; and Lemmas 4.5 and 5.1 to 5.3 and Corollary 4.6 hold for $A_U(e, a, b, B)$ with $\pi_\circ^e$ in place of $\pi_\circ$ and $E_*$ included in $\mathcal{E}(t)$. The states and letters of $A_U(e, a, b, B)$ do not depend on $e$, $a$, $b$ or $B$.*

*Proof.* The equation $E_*$ holds if and only if $n_{0,0} = e$; together with the modified (W2) this says that the certificate starts at $(\mathsf{start}, (e, 0, 0, 0))$, and the rest of the proof of Lemma 4.3 is unchanged. The parser modified by (I1) satisfies Lemma 4.4 with the modified (W2), by the same proof; in particular, after a row with head $\dashv$ only # is allowed. In the proof of Lemma 4.5 the factors contributed by $E_*$ were computed above, and the choice for $E_*$ is independent of all other choices; the other factors are as before. The proofs of Corollary 4.6 and Lemmas 5.1 to 5.3 use only Lemmas 4.1 and 4.5, the fact that the probability that is not absorbed is redistributed according to the initial distribution, and the parser, and they carry over. The parser depends only on $U$; the state set consists of parser states, the marks $\mathsf{j}, \mathsf{s}, \dagger$, the slot vectors $\xi$ for four counters, the register and the two absorbing states; the alphabet is $\Sigma_U$. The parameters $e, a, b, B$ enter only the transition probabilities and the initial distribution. □

**Theorem 7.3.** *Let $N_0$ be the number of states of $A_U(e, a, b, B)$, which does not depend on $e, a, b, B$ by Lemma 7.2, let $l$ be the least integer with $2^l \geq |\Sigma_U|$, and let $N = N_0(2^l - 1) + 1$. Every left-c.e. real in $[0, 1]$ is the value of a rational probabilistic finite automaton over $\{0, 1\}$ with at most $N$ states. In particular Theorem B holds.*

*Proof.* The values 0 and 1 are realized by one-state automata over $\{0, 1\}$. Let $\alpha \in (0, 1)$ be left-c.e., choose $a, b, B$ as in Theorem 5.4, and choose $e$ with $W_e = W_\alpha$ by Lemma 3.3. By Lemma 7.2, the proof of Theorem 5.4 applies verbatim to $A_U(e, a, b, B)$, with $L(U, (e, 0, 0, 0)) = W_e = W_\alpha$ in place of $L(P)$; so its value is $\alpha$. Lemma 7.1 turns it into an automaton over $\{0, 1\}$ with $N$ states and the same value. □

# 8 Uniformity

**Proposition 8.1.**

*(i) Whether $V(A) = 0$ is decidable for rational probabilistic finite automata $A$.*

*(ii) There is no algorithm that, given a left-c.e. index of a real $\alpha \in [0, 1]$, outputs a rational probabilistic finite automaton with value $\alpha$. This remains true for indices of sequences whose limits lie in $\{0, 1/2\}$.*

*(iii) There is an algorithm that, given a left-c.e. index of a real $\alpha$ and rationals $a, b$ with $0 < a < \alpha < b < 1$, outputs a rational probabilistic finite automaton over $\{0, 1\}$ with at most $N$ states and value $\alpha$.*

*Proof.* (i) $P_A(w) > 0$ if and only if some path of states, starting in the support of $\pi$ and ending in $F$, has positive probability under $w$. So $V(A) > 0$ if and only if $F$ is reachable from the support of $\pi$ in the directed graph with an edge $p \to p'$ whenever $M_\ell(p, p') > 0$ for some letter $\ell$, which is decidable.

(ii) For each $n$, the sequence $\alpha_t^{(n)}$ equal to $1/2$ if the Turing machine $M_n$ of Section 6 halts on input $n$ within $t$ steps, and to 0 otherwise, is computable uniformly in $n$, nondecreasing, and has limit $1/2$ if $n \in H$ and 0 otherwise. An algorithm as in (ii), followed by the test of (i), would decide $H$.

(iii) From $a$ and $b$ compute the least $B$ with (1). From the index of $\alpha$ and from $a, b$ compute an enumeration index of $W_\alpha$, then $e$ with $W_e = W_\alpha$ (Lemma 3.3), and output the automaton of Theorem 7.3. □

The map from automata to left-c.e. indices of their values is computable (Proposition 2.3). By (ii) and (iii), a left-c.e. index of the target does not suffice to compute an automaton, while

an index together with a rational interval $(a, b)$ does. The obstruction in (ii) concerns the lower end; we do not know whether an index together with a rational $a \in (0, \alpha)$ alone suffices.

# 9 Quantum automata: where the ladder breaks

Through the folklore simulation of probabilistic automata by quantum automata, Theorem A determines the set of values of quantum models that keep classical records of their measurements; measure-once automata with algebraic data, which evolve coherently until one final measurement, have algebraic values. The one-way Kondacs–Watrous measure-many model measures after every letter and continues on a single outcome; Theorem 11.5 proves that it too has the full left-c.e. spectrum. Throughout this section all finite data, including the initial state and measurement operators, are algebraic unless explicitly required to be rational. The value is the supremum of the acceptance probabilities of finite input words. A rational complex matrix means one whose real and imaginary parts are rational.

## 9.1 The two ends

**Proposition 9.1** (measure-once values; implicit in [9])**.** *Every measure-once quantum finite automaton with algebraic data has an algebraic value, uniformly computable from its data.*

*Proof.* Write its acceptance probability as $\psi^* U_w^* E U_w \psi$, where the letter matrices are unitary and $0 \le E \le I$ is the final accepting effect; fixed endmarker operations can be absorbed into $\psi$ and $E$. The closure $G$ of the letter monoid is a compact group. Derksen, Jeandel and Koiran note [9, Section 2.2], by a theorem of Onishchik and Vinberg, that this compact closure is algebraic, and give an algorithm [9, Section 3] computing polynomial equations that define it; they work with real orthogonal matrices. Complex unitaries are handled by realification. Hence

$$V = \max_{g \in G} \psi^* g^* E g \psi$$

is the endpoint of an effectively semialgebraic subset of the real line. Real quantifier elimination makes it algebraic and computes it. Algebraicity is thus immediate from [9, Section 2.2], whose decision procedure evaluates exactly such a first-order sentence; it sharpens the computability that follows from the strict-threshold decision theorem of [4]. We claim no priority for it. □

Here a *rational quantum channel* is a completely positive trace-preserving (CPTP) linear map whose superoperator matrix in the matrix-unit basis has rational complex entries. This condition is on the channel, and does not require a Kraus representation with rational entries.

**Proposition 9.2** (channel values)**.** *The values of automata whose letters act by rational quantum channels, with a rational initial density matrix and a rational final accepting effect, are exactly the left-c.e. reals in* $[0, 1]$.

*Proof.* On a finite word the final density matrix and its acceptance probability are computable (in fact rational), so enumeration proves the upper inclusion. For the converse, embed a row-stochastic matrix $M$ as

$$\Phi_M(\rho) = \sum_{i,j} M(i,j) \rho_{ii} \, |j\rangle\langle j|.$$

This is a rational CPTP map: Kraus operators $\sqrt{M(i,j)}\,|j\rangle\langle i|$ witness complete positivity and trace preservation. Starting at $\mathrm{diag}(\pi)$ and using effect $\mathrm{diag}(\mathbf{1}_F)$ gives exactly the PFA's word

probabilities. Apply Theorem A. This classical embedding is the folklore exact simulation recorded in [1, Theorem 4.2]. □

## 9.2 Quantum automata with control language

Bertoni, Mereghetti and Palano introduced *quantum finite automata with control language* (QFA-CL) [3, 21]. After each letter and a right endmarker, the machine applies the corresponding unitary and measures a fixed observable. All outcomes continue; acceptance means that the complete outcome sequence belongs to a fixed regular language. The classical recognizer reads outcomes but does not change the quantum operations. We use the special case of complete rank-one basis measurement. This differs from the Kondacs–Watrous measure-many model, where acceptance and rejection halt and there is only one continuing measurement outcome.

The exact PFA simulation by general quantum automata is folklore [1, Theorem 4.2]; that survey also lists QFA-CL among the equivalent general models. The following value statement is a corollary of that simulation and Theorem A. We give a rational-unitary realization explicitly.

**Corollary 9.3** (control-language values)**.** *The values of QFA-CL with algebraic data are exactly the left-c.e. reals in* $[0,1]$*. Rational real unitaries, a basis initial state and complete basis measurements already suffice.*

*Proof.* For a fixed word there are finitely many outcome sequences, with computable algebraic probabilities. Summing the probabilities of those in the control language and enumerating words proves the upper inclusion.

For the lower inclusion, first give a rational PFA a deterministic start without changing its value: add a fresh nonaccepting start state and an opening letter which initializes its former initial distribution; old letters from the new start, and the opening letter from old states, go to a nonaccepting absorbing sink. Only a word with exactly one opening letter, at its beginning, can accept.

For a dyadic PFA choose a common transition denominator $D = 4^m$. For each letter $a$ partition $\{0, \dots, D-1\}$ in each state $q$ so that a deterministic update $\delta(q, a, j)$ selects $q'$ for exactly $DM_a(q, q')$ choices of $j$. Number the $k$ input letters $1, \dots, k$. On $\mathbb{C}^{k+1} \otimes \mathbb{C}^D$ use

$$U_a = S^a \otimes W_D,$$

where $S$ is cyclic shift and $W_D$ is the Walsh matrix with entries $\pm 2^{-m}$. Start in $|0, 0\rangle$, measure the basis after each step, and use identity at the endmarker. Each ordinary outcome $(g, j)$ supplies a fresh uniform $j$; the difference of successive clock coordinates $g$, modulo $k+1$, identifies the input letter. Difference zero identifies the endmarker. A finite control recognizer stores the previous clock coordinate and the simulated PFA state, updates by $\delta$, and tests the accepting set at the endmarker. It rejects any outcome after that test. Induction gives equality of every word probability, including the empty word. Every unitary used here is rational.

For a general rational PFA with common denominator $D > 1$, put $r = \lceil \log_2 D \rceil$. Use a four-dimensional Walsh coin $W_4$ and a sufficiently large cyclic clock to distinguish the input types $a$, $c$, $d$ and the endmarker exactly as above. Encode a PFA word $a_1 \cdots a_h$ by blocks

$$a_1 c^{L_1} d \; a_2 c^{L_2} d \; \cdots \; a_h c^{L_h} d.$$

The recognizer retains $a_i$ and the current PFA state. Each $c$ supplies one fair bit from its uniform four-way outcome. Groups of $r$ bits give independent uniform integers $j < 2^r$. A

trial with $j \geq D$ is discarded; the first $j < D$ makes the update $\delta(q, a_i, j)$ and sets a success flag. Further coins in that block are ignored. The delimiter rejects unless the flag is set. Partial trials are discarded at the delimiter. An endmarker is allowed only between blocks; malformed inputs reject. This uses finitely many classical states and no quantum feed-forward.

The success probability of a block is

$$s(L) = 1 - (1 - D/2^r)^{\lfloor L/r \rfloor}.$$

Conditional on success its update is exactly $M_a$, independently of the trial of first success. The acceptance probability of the displayed code is consequently

$$P_A(a_1 \cdots a_h) \prod_{i=1}^{h} s(L_i). \tag{2}$$

It is at most $V(A)$, and for each fixed word tends to its PFA acceptance probability as all $L_i \to \infty$. The empty code implements the empty word. If $D = 1$, update deterministically on $a$ and set the success flag immediately. Thus values are preserved for every rational PFA, using only rational Walsh matrices and permutation clocks. Theorem A completes the proof. □

### 9.3 The one-way measure-many model

For one-way Kondacs–Watrous (KW) measure-many automata [17], a convenient normal form, obtained by restricting each letter unitary to the nonhalting space, consists of continuing contractions $T_a$ (the nonhalting block), accepting and rejecting effects $Q_a, R_a \geq 0$ (the Gram matrices of the accepting and rejecting blocks) with

$$T_a^* T_a + Q_a + R_a = I,$$

a unit initial live vector $\psi$, and a final accepting effect $0 \leq E \leq I$. Write $T_w = T_{a_h} \cdots T_{a_1}$ and

$$A_w = \sum_{i=1}^{h} T_{a_1 \cdots a_{i-1}}^* Q_{a_i} T_{a_1 \cdots a_{i-1}}, \qquad f(w) = \psi^*(A_w + T_w^* E T_w)\psi.$$

Here $T_\varepsilon = I$ and $A_\varepsilon = 0$. In particular $0 \leq A_w \leq I - T_w^* T_w$. A fixed left-endmarker merely adds an algebraic initial acceptance and leaves an algebraic live vector of norm at most one; the conclusions below extend by this affine normalization. Every $f(w)$ is algebraic and computable, so the value is left-c.e.

**Question 9.4.** Is the value of every one-way KW measure-many automaton with algebraic data a computable real? Is there a uniform algorithm to approximate these values from the automata?

Both questions, retained here with their version 1.1.0 label, are answered negatively by Theorem 11.5. The following two subclasses from that version still illustrate where effective upper bounds can be obtained.

**Proposition 9.5** (unitary or rank at most one)**.** *If every ordinary continuing operator $T_a$ is globally unitary or has rank at most one, then the value is algebraic and uniformly computable.*

*Proof.* Let $G$ be the compact closure of the unitary-letter monoid; these letters have $Q_a = R_a = 0$. For each nonzero rank-one letter choose an algebraic unit vector $v_a$ spanning its image, and

put $v_0 = \psi$. At the beginning of a unitary stretch the live vector is a scalar multiple of one of these finitely many vectors. The stretch acts as $g \in G$. For mode $b$ set

$$e_b(g) = (gv_b)^* E g v_b, \quad q_{ba}(g) = (gv_b)^* Q_a g v_b, \quad s_{ba}(g) = \|T_a g v_b\|^2.$$

A nonzero rank-one letter earns $q_{ba}(g)$ and leaves mass $s_{ba}(g)$ in mode $a$; a rank-zero letter earns $q_{ba}(g)$ and leaves no live mass. Stopping after the stretch earns $e_b(g)$. These functions are algebraic polynomials in the real coordinates of $g$, with $q_{ba} + s_{ba} \le 1$.

Write $\mathcal{B}$ for the finite set of modes, including mode 0. Let $\mathcal{P} \subseteq [0,1]^{\mathcal{B}}$ consist of the vectors $z$ satisfying, for all modes $b$ and all $g \in G$,

$$z_b \ge e_b(g), \qquad z_b \ge q_{ba}(g) + s_{ba}(g) z_a$$

for nonzero rank-one letters, and $z_b \ge q_{ba}(g)$ for rank-zero letters. The all-ones vector is feasible. Let $(\Theta z)_b$ be the maximum over $g \in G$ of $e_b(g)$, of $q_{ba}(g) + s_{ba}(g) z_a$ over nonzero rank-one letters $a$, and of $q_{ba}(g)$ over rank-zero letters $a$; thus $\mathcal{P} = \{z \in [0,1]^{\mathcal{B}} : z \ge \Theta z,\ z_b \ge \max_G e_b\}$. Finite-horizon Bellman iteration from $\max_G e_b$ increases to a vector $F$. The Bellman map $\Theta$ is monotone and 1-Lipschitz, so $F$ is a fixed point and is the componentwise least member of $\mathcal{P}$. Every finite sequence of closure choices is approximable by actual unitary stretches, by continuity; hence $V = F_0 = \min_{z \in \mathcal{P}} z_0$. This argument uses neither eventual halting nor uniqueness of a fixed point.

The effective algebraic description of $G$ is due to Derksen, Jeandel and Koiran [9]. Real quantifier elimination now describes $\mathcal{P}$ and its minimum coordinate. Thus that coordinate is algebraic and computable. With no unitary letters take $G = \{I\}$; with no nonunitary letters this is precisely the measure-once maximization of Proposition 9.1. □

We give the effective compactness argument needed for the second subclass, including its regular-language step.

**Lemma 9.6** (regular unitary products)**.** *Given algebraic unitary letters and a regular language $L$, the closure of their products over $L$ is effectively compact semialgebraic. If $L$ is nonempty, finite $\epsilon$-nets made of actual $L$-words can be computed for every rational $\epsilon > 0$.*

*Proof.* In a strongly connected component of a finite recognizer choose a root and internal paths from the root to each state $q$ and back, with products $A_q, B_q$. Products are chronological, with later factors on the left. Let $H$ be the compact group generated by the root loops $B_q A_q$ and $B_{q'} U_a A_q$ for internal edges $q \xrightarrow{a} q'$. Since

$$A_{q'}^{-1} U_a A_q = (B_{q'} A_{q'})^{-1} (B_{q'} U_a A_q) \in H,$$

corrected edges telescope. The closure of internal $p$-to-$q$ path products is exactly $A_q H A_p^{-1}$. The reverse inclusion uses actual paths going first from $p$ to the root, then around root loops, then to $q$, together with $B_p A_p \in H$. The closure of a unitary monoid is a group, so inverses in these expressions are legitimate limits of positive products.

A path in the recognizer traverses each strongly connected component at most once. There are finitely many choices of traversed components, connecting edges and entry and exit states. Thus its accepted-product closure is a finite union of finite products of these compact cosets and fixed matrices. Each $H$ is effectively algebraic by [9]; quantifier elimination describes the products and unions. Emptiness of $L$ is decidable. If it is nonempty, enumerate actual $L$-words and test whether their open $\epsilon$-balls cover the compact closure, a semialgebraic condition. Density and compactness guarantee that this search terminates. □

**Proposition 9.7** (common upper block-triangular form)**.** *Suppose the continuing operators admit a common orthogonal decomposition $\mathbb{C}^d = H_1 \oplus \cdots \oplus H_r$ of the live space in which they are upper block triangular, and each diagonal letter block is either unitary or strictly contracting in operator norm. Then their value is uniformly computable. The decomposition may be supplied or found effectively; existence of such a decomposition is decidable. These conclusions also hold with a regular restriction on the input words and an end effect depending on the final state of a finite recognizer.*

*Proof.* A unitary diagonal block of a contraction forces all other blocks in its column to vanish: the diagonal image already has full norm. Applying the same argument to the adjoint also clears its row. For a subalphabet $S$, let $K_S$ be the direct sum of the blocks on which every $S$-letter is unitary. It reduces every $S$-letter; all $Q_a, R_a$ for $a \in S$ annihilate it.

On $K_S^\perp$ there are $r' \le r$ remaining blocks, each strictly contracted by some $S$-letter. A word containing every $S$-letter has each diagonal block norm at most

$$\rho = \max_j \prod_{a \in S} \|T_{a,jj}\| < 1.$$

Choose an effective rational $0 < \rho_1 < 1$ above $\rho$. A product of $m$ such complete words has at most $r' - 1$ strict block-index changes along each monotone block path, so its norm on $K_S^\perp$ is bounded by

$$r' \sum_{j=0}^{\min(r'-1,m)} \binom{m}{j} (r')^j \rho_1^{m-j}. \tag{3}$$

Off-diagonal factors have norm at most one. This explicit bound tends to zero and supplies an effective $N_S$ complete blocks giving norm at most $1/2$. If $K_S^\perp = 0$, take $N_S = 1$. Grouping blocks gives norm at most $2^{-k}$ after $kN_S$ complete blocks.

For completeness we explain how this gives effective finite nets of actual word instruments $J_w = (T_w, A_w)$, in the metric $\max(\|\Delta T\|, \|\Delta A\|)$, preserving any prescribed finite-monoid image of $w$. Replacing one segment by a $\delta$-close instrument changes the whole instrument by at most $3\delta$, since all continuing operators and accumulated effects have norm at most one. Proceed by induction on $|S|$, starting with the empty word. Fix $\eta > 0$, choose $k$ so that $2^{1-k}$ and $2^{-2k}$ are below $\eta/4$, and put $t = kN_S$. Use greedy minimal $S$-complete blocks.

If a word has at least $t$ complete blocks, write it $ps$ where $p$ ends at the $t$-th block. By Lemma 9.6, replace $s$ by a representative $u$ in the same finite-monoid fiber, whose action on $K_S$ is within $\eta/4$ of that of $s$. Finitely many such representatives cover all fibers. Because $p$ contracts the complement,

$$\|T_{ps} - T_{pu}\| \le \max(\eta/4, 2^{1-k}), \qquad \|A_{ps} - A_{pu}\| \le 2^{-2k}.$$

The second estimate is $A_{ps} - A_{pu} = T_p^*(A_s - A_u)T_p$, using that both suffix effects vanish on $K_S$. A minimal complete block has form $va$ with $v$ over $S \setminus \{a\}$. Replace each of the $t$ such proper-subalphabet stretches in $p$ by its inductively computed net representative, in the same monoid fiber. If there are fewer than $t$ complete blocks, apply the same replacement and also replace the incomplete final stretch, which uses a proper subalphabet. There are at most $t+1$ replaced stretches. Accuracy $\eta/[6(t+1)]$ for each makes their total contribution at most $\eta/2$. Crucially the long suffix is replaced first, before approximating $p$.

There are finitely many resulting templates and choices of representatives, all effectively enumerable; they form an $\eta$-net of actual-word instruments in each monoid fiber. For a regular restriction use the recognizer's transition monoid, retaining its final state and end effect. Changing $J_w$ by $\eta$ changes its final acceptance by at most $3\eta$. Thus the maximum

$L_\eta$ over the permitted net words satisfies $L_\eta \le V \le L_\eta + 3\eta$, which computes $V$. Empty permitted languages are decidable and may be assigned value zero.

Finally enumerate the finitely many ordered partitions of $d$ and quantify over a unitary change of basis. Vanishing lower blocks and, on each diagonal block $B$, the disjunction $B^*B = I$ or $I - B^*B$ positive definite are first-order real algebraic conditions. Quantifier elimination decides existence and supplies algebraic data for a decomposition. Scalar blocks show that simultaneous unitary triangularization, and in particular commuting continuing operators, are included. □

**Scope of the ladder.** Propositions 9.1 and 9.2, Corollary 9.3, and Theorem 11.5 give algebraic measure-once values and the full left-c.e. spectrum for rational channels, QFA-CL and one-way KW automata. A KW automaton is a QFA-CL using the control language $c^*a\{a, r, c\}^*$, where $a, r, c$ are the accept, reject and continue outcomes; a measure-once automaton is KW with halting only at the endmarker. No coherent head position or two-way motion is used in our new construction. Our isolation results apply to KW and models containing its acceptance functions, including NaQFA and BiQFA in the terminology of [1]. They do not settle the Latvian model.

### 9.4 The Latvian boundary

Latvian letters are unitary operations followed by unread projective measurements, or finite compositions of these; their channels are unital and trace preserving. All outcomes continue, with acceptance only at the end. This differs from KW's halting measurement. The following short boundary result does not decide Latvian values or isolation.

**Proposition 9.8** (unital boundary bound)**.** *Let $\Phi$ be a unital trace-preserving completely positive map on $D$-dimensional matrices, $\rho$ a density matrix and $0 \le E \le I$. Put $f_n = \operatorname{tr}(E\Phi^n(\rho))$. The Cesàro state $\sigma = \lim_N N^{-1} \sum_{n=1}^N \Phi^n(\rho)$ exists and*

$$\rho \le \operatorname{rank}(\sigma)\sigma, \qquad 1 - f_0 \le \operatorname{rank}(\sigma) \lim_N \frac{1}{N} \sum_{n=1}^{N} (1 - f_n).$$

*Consequently the least dimension of a Latvian automaton satisfying $f(\varepsilon) = 0$ and $f(a^n) \ge 1 - \varepsilon$ for all $n \ge 1$, for $0 < \varepsilon \le 1$, is exactly $\lceil 1/\varepsilon \rceil$, allowing algebraic data. In particular the exact unary zero/one nonempty-word acceptance function cannot be preserved in finite Latvian dimension.*

*Proof.* The Schwarz inequality, unitality and trace preservation make $\Phi$ a Hilbert–Schmidt contraction. For a contraction on a finite-dimensional Hilbert space, its fixed space equals its adjoint's fixed space: equality in the norm bound for a fixed vector gives $\Phi^*\Phi x = x$, hence $\Phi^* x = x$, and conversely. The orthogonal complement is $\operatorname{range}(I - \Phi)$. Cesàro averages converge on it by telescoping, proving existence of $\sigma$ and $\Phi(\sigma) = \sigma$.

Write $\Phi(X) = \sum_j K_j X K_j^*$. Both $\sum K_j^* K_j$ and $\sum K_j K_j^*$ equal $I$. Expanding squared commutators gives

$$\sum_j \|K_j\sigma - \sigma K_j\|_F^2 = 2\operatorname{tr}(\sigma^2) - 2\operatorname{tr}(\sigma\Phi(\sigma)) = 0.$$

Every Kraus operator therefore commutes with $\sigma$, its kernel projection $P$ and pseudoinverse $H = \sigma^+$. These last two matrices are fixed by $\Phi^*$. Pairing with the Cesàro limit gives $\operatorname{tr}(P\rho) = \operatorname{tr}(P\sigma) = 0$, so positivity supports $\rho$ on $\operatorname{supp}\sigma$. Likewise $\operatorname{tr}(H\rho) = \operatorname{tr}(H\sigma) = r = \operatorname{rank}\sigma$. On this support, $Z = \sigma^{-1/2}\rho\sigma^{-1/2}$ is positive with trace $r$, hence $Z \le rI$. This proves $\rho \le r\sigma$. Pair

with $I - E$ to get the displayed probability inequality. A fixed left marker is absorbed into $\rho$ and a right-marker channel into $E$ by its adjoint.

If every $f_n \geq 1 - \varepsilon$ for $n \geq 1$, the inequality gives $f_0 \geq 1 - D\varepsilon$, so $f_0 = 0$ requires $D \geq 1/\varepsilon$. Conversely start on the first basis state, apply the $D$-dimensional Fourier matrix then complete basis dephasing at each letter, and finally accept all states except the first. The first letter produces $I/D$, which stays fixed; therefore $f_0 = 0$ and $f_n = 1 - 1/D$ for every $n \geq 1$. Taking $D = \lceil 1/\varepsilon \rceil$ is sharp. Fourier entries are algebraic. If $D$ is a perfect square, a rational Householder reflection sending the first basis vector to the rational uniform unit vector gives the same sharp example with real rational gates. □

The fixed-point commutant structure is due to Kribs [39, Theorem 2.1]; the $D$-version of the domination inequality is known in substance from the finite-dimensional Pimsner–Popa index formula, as presented by Gao and Rouzé [38, Section 2.2, (17)–(18)]. The rank argument above is elementary. The saturating Fourier-and-dephasing automaton is already the base case $M^{(1)}$ of Mereghetti, Palano and Raucci [40, Section 3], adapted there from Ambainis et al. [32]; we claim no novelty for that example or the operator structure. The automaton lower bound supplies the matching least dimension. General Latvian value computability and both isolation decisions remain open here.

## 10 Open problems

**Small automata.** The automata of Theorem B have a fixed but large number of states. Few states are not enough.

**Proposition 10.1.** *The value of a rational probabilistic finite automaton with at most two states is rational.*

*Proof.* With one state the value is 0 or 1. With states $0, 1$, let $x$ be the probability of state 1. Each letter acts by an affine map $f_\ell(x) = M_\ell(0,1) + (M_\ell(1,1) - M_\ell(0,1))x$ with rational coefficients, mapping $[0,1]$ into itself. Let $O$ be the set of the numbers $f_w(x_0)$ over all words $w$, where $x_0$ is the initial probability of state 1, and let $x_- = \inf O$ and $x_+ = \sup O$. Since the maps $f_\ell$ are affine and continuous and $f_\ell(O) \subseteq O$, the interval $[x_-, x_+]$ is invariant under every $f_\ell$; and every closed interval that contains $x_0$ and is invariant under every $f_\ell$ contains $O$, hence $[x_-, x_+]$. An interval $[u, v]$ is invariant under an affine map $f$ if and only if $f(u), f(v) \in [u, v]$. So the pairs $(u, v)$ with $0 \leq u \leq x_0 \leq v \leq 1$ and $u \leq f_\ell(u), f_\ell(v) \leq v$ for all $\ell$ form a rational polytope that contains $(x_-, x_+)$ and on which $v \geq x_+$ and $u \leq x_-$. The minimum of $v$ on this polytope is $x_+$, and it is attained at a vertex, which is rational; likewise $x_-$ is rational. The value is $x_+$, $1 - x_-$, 0 or 1, according to the accepting set. □

**Question 10.2.** Is every left-c.e. real in $[0,1]$ the value of a rational probabilistic finite automaton with three states and two letters? Less ambitiously, give an explicit small number of states, or an explicit automaton with few states and a noncomputable value.

**Fixed matrices.** For $\alpha \in (1/8, 1/2)$ the constants $a = 1/8$, $b = 1/2$ and $B = 16$ can be kept fixed in Theorem B, and then $e$ enters only through the initial distribution $\pi_\circ^e$, which is also used in the rows (P3). Rote's fixed automaton has the initial distribution as its only input [28]. This suggests the following question.

**Question 10.3.** Are there finitely many fixed rational stochastic matrices and a fixed accepting set such that, as the rational initial distribution varies, the values include every left-c.e. real in some nondegenerate interval?

The answer to this retained question is yes, even for the whole interval $[0,1]$; see Theorem 12.2. Its proof changes the reset rule so that the transition rows are genuinely independent of the initial distribution.

**Noise.** The construction uses transitions of probability 0: acceptance and rejection are permanent, and a failed test stays dead until the next delimiter. Some transitions of probability 0 are needed for a noncomputable value.

**Proposition 10.4.** *If all entries of all matrices $M_\ell$ are positive, then $V(A)$ is computable from $A$.*

*Proof.* Let $n = |Q|$, let $m_0 > 0$ be the least entry of all $M_\ell$, and put $\eta = nm_0 \le 1$. Write $M_\ell = \eta J + (1-\eta)S_\ell$, where $J$ has all entries $1/n$; if $\eta < 1$, each $S_\ell$ is stochastic, and if $\eta = 1$, every $M_\ell$ equals $J$. For probability vectors $\nu, \nu'$ the difference $\nu - \nu'$ has coordinate sum 0, so $(\nu - \nu')J = 0$ and $\|(\nu - \nu')M_\ell\|_1 \le (1-\eta)\|\nu - \nu'\|_1$. For a word $w = w'v$ with $|v| = k$, applying this to $\nu = \pi M_{w'}$ and $\nu' = \pi$ gives $|P_A(w) - P_A(v)| \le \frac{1}{2}\|(\nu - \nu')M_v\|_1 \le (1-\eta)^k$. Hence $V(A)$ lies between the maximum of $P_A$ over words of length at most $k$ and that maximum plus $(1-\eta)^k$, for every $k$. □

**Other finite-state models.** A partially observable Markov decision process with finitely many states, actions and observations, rational transition probabilities, a rational initial distribution and a deterministic observation map, together with a target set, has as its value the supremum, over all policies that may use the history of actions and observations and may randomize, of the probability of ever visiting the target.

**Proposition 10.5.** *The values of such processes with reachability objectives are exactly the left-c.e. reals in $[0,1]$.*

*Proof.* Let $V_k$ be the supremum over policies of the probability of visiting the target within $k$ steps. For a fixed horizon, this probability is affine in each of the distributions that a policy chooses after the histories of length less than $k$, separately, since each history occurs at most once along a trajectory. Replacing these distributions one at a time by suitable single actions therefore does not decrease it, so $V_k$ is attained by one of finitely many deterministic policies and is a rational number computable from $k$. The sequence $V_k$ is nondecreasing, $V_k$ is at most the value, and for every policy the probability of ever visiting the target is the limit of the probabilities of visiting it within $k$ steps, hence at most $\sup_k V_k$. So the value is $\lim_k V_k$, a left-c.e. real in $[0,1]$. Conversely, let $\alpha \in (0,1)$ be left-c.e. and let $A$ be an automaton with value $\alpha$ from Theorem 5.4; view it as a process with one observation, actions $\Sigma_P$ and target $\{\mathsf{yes}\}$. Since yes is absorbing, the probability of visiting yes within $k$ steps under a sequence of actions is the acceptance probability of its first $k$ letters, so every policy visits yes with probability at most $\alpha$, and the policies that play $t^m$ show that the value is $\alpha$. The values 0 and 1 are trivial. □

A probabilistic Büchi automaton [2] accepts an infinite word with the probability that the run visits the accepting set infinitely often. For the automata of Theorem 5.4, whose only accepting state is absorbing, this is the probability of ever reaching yes, so the supremum of these probabilities over infinite words is again $\alpha$. Thus every left-c.e. real in $[0,1]$ is such a supremum for a probabilistic Büchi automaton.

**Question 10.6.** Which reals are suprema of the acceptance probabilities of rational probabilistic Büchi automata? Are they all left-c.e.?

Elkouss and Pérez-García showed, through probabilistic automata, that the capacity of a finite-state channel cannot be computed from the channel to within a fixed precision [11], and Boche, Schaefer and Poor proved further non-computability results of this uniform kind [5]. They treat a single computable channel with a noncomputable capacity as a stronger form of non-computability [5, Remark 31]. Such channels exist with finitely many states.

**Corollary 10.7.** *Every left-c.e. real in* $[0,1]$ *is the capacity of a finite-state channel with rational transition probabilities. In particular, some finite-state channel with rational transition probabilities has a capacity that is not a computable real.*

*Proof.* Elkouss and Pérez-García map each probabilistic finite automaton $A$ that starts in a fixed state to a finite-state channel $V_A$ with rational transition probabilities, in which one input register controls $A$, with identity and reset controls added, and they prove that the capacity of $V_A$ equals the value of $A$ [11, Theorem 1]. An automaton with a rational initial distribution $\pi$ is converted to one with a fixed start state without changing its value: add a start state $s$, a nonaccepting absorbing sink and a new letter that sends $s$ to $\pi$ and every other state to the sink, and let each old letter act as before on the old states and send $s$ to the sink. Only words that begin with the new letter can be accepted, and the probability of accepting such a word is the probability that the old automaton accepts the rest of it. Now apply Theorem 5.4. Theorem C gives a channel with capacity $\alpha_*$, and Corollary D gives channels whose capacities are halting probabilities. □

The corollary is an inclusion; we do not determine which reals are capacities of finite-state channels with rational transition probabilities.

Measure-once automata, quantum channels, quantum automata with control language and one-way Kondacs–Watrous measure-many automata are compared in Section 9.

# 11 The one-way measure-many spectrum

We work over real rational coordinates for the lower construction. All vectors below are unnormalized live vectors: a rejected squared norm is probability already lost, and is never divided out. A phase is an orthogonal coordinate block in one finite Hilbert space, not an observed classical state. The right endmarker rejects all residual live mass.

## 11.1 A projection compiler

**Lemma 11.1** (rational completion and binary control)**.** *Suppose finitely many logical letters have rational continuing maps $T$ and accepting and rejecting effects $Q, R$, where $T^tT, Q, R$ are rational orthogonal projections with sum $I$. They are realizable by rational orthogonal KW gates with three fixed coordinate outcomes, in total dimension three times the live dimension. With $m$ logical letters, $m$ tagged live copies give an exactly value-preserving binary realization. Its continuing blocks are a finite-order permutation and a partial isometry.*

*Proof.* Stack $T, Q, R$ in three output blocks. The resulting columns are rational and orthonormal. Any rational orthonormal frame extends rationally to an orthogonal matrix: after earlier

columns have been sent to the corresponding coordinate vectors, send the next unit column $v$ to $e_j$ by the reflection

$$I - 2\frac{(v-e_j)(v-e_j)^t}{\|v-e_j\|^2},$$

or the identity if $v = e_j$. This reflection fixes earlier columns. Inverting the product completes the original frame. The three output blocks are measured as continue, accept and reject. A separate right gate swaps live and rejecting coordinates.

For binary control, letter 0 cycles the $m$ tagged copies. Letter 1 applies logical letter $j$ on copy $j$, retaining its copy label on continuation and using orthogonal outcome blocks for different copies. Start in copy zero. Every binary word decodes to the successive logical letters named by the copy at each 1; trailing 0's do not accept. Conversely, cyclic shifts reach every desired next letter. Thus the word probabilities, and their suprema, agree. The 0 block is a permutation and the 1 block is a direct sum of partial isometries. Apply the preceding completion once to these binary blocks. □

Fix a finite directed graph with distinct start and terminal vertices. The terminal is reachable and has no outgoing edge except a padding loop $\gamma I$. Each edge $e$ carries a rational $d \times d$ matrix $A_e$ of norm at most a rational $0 < \gamma < 1$. Fix $0 \neq \xi \in \mathbb{Q}^d$ and a rational unit vector $u$. Products are chronological, last factor on the left. For start-to-terminal paths set

$$L_w = \operatorname{span}(A_w\xi), \qquad p = \sup_w \|P_{L_w}u\|^2,$$

with $P_{\{0\}} = 0$. Scalar padding does not change any nonzero ray.

**Theorem 11.2** (graph projections)**.** *From these data one effectively constructs a rational KW automaton with basis start and value*

$$V = \gamma^4 + (1-\gamma^4)p. \tag{4}$$

*Its size is bounded in terms of the graph size and d, independently of the rational coefficients. It can be binary with the continuing-block forms of Lemma 11.1.*

*Proof.* The reset phase has coordinates $(r, x) \in \mathbb{R} \oplus \mathbb{R}^d$, initially on the basis reservoir axis $r$. At reset allow plane rotations and a start letter which projects $x$ onto $\operatorname{span}\xi$, retains $r$, and moves to the start vertex. At a vertex the coordinates are again $(r, x)$. Selecting edge $e$ embeds them as $(r, x, 0)$ in a write phase $(r, x, y)$. Allow rotations on $(r, y)$, protecting $x$. The check projects $(x, y)$ onto the graph of $A = A_e$ and moves to an erase phase. There allow rotations on $(r, x)$, protecting $y$. Commit rejects the old $x$ and relabels $y$ as the register at the destination. A terminal read accepts $uu^t x$, rejects its orthogonal complement in $x$, and continues $r$ alone to reset. No other operation accepts. Every letter rejects every phase other than its specified source. Consequently wrong-phase words and incomplete final rounds cannot add acceptance.

The graph projector is rational:

$$C(C^tC)^{-1}C^t, \qquad C = \begin{pmatrix} I \\ A \end{pmatrix}.$$

Each continuing block is an isometry after a projection, and its accepting, rejecting and continuing effects are orthogonal rational projections. Thus Lemma 11.1 realizes the specified instrument with ordinary KW gates.

For each allowed plane use the single rotation with cosine $3/5$ and sine $4/5$. Its positive powers are dense in the plane rotation group. Indeed if $(3+4i)/5$ were a root of unity, its sum with its inverse, $6/5$, would be a rational algebraic integer, an impossibility. An irrational rotation has dense powers: the pigeonhole principle supplies nonzero multiples arbitrarily close to zero modulo a turn, and successive multiples of such a small angle give arbitrarily fine circle nets. Plane rotations generate the norm-preserving changes of a vector in each allowed coordinate space by successive elimination of coordinates. A finite sequence of ideal rotations is therefore approximable by finite words in our actual rational letters. We use this only for lower bounds, by continuity of finite products and squared norms. Upper bounds allow arbitrary rotations.

At one check write the unique orthogonal decomposition

$$\begin{aligned} (x,y) &= (v, Av) + (-A^t h, h), \\ v &= x + A^t h, \qquad y = Ax + (I + AA^t)h, \\ x_{\text{next}} &= Ax + AA^t h. \end{aligned} \tag{5}$$

The rejection there is $e^2 = \|h\|^2 + \|A^t h\|^2$. Put

$$f = \frac{\gamma^2}{\sqrt{1+\gamma^2}}, \qquad S = \frac{f^2}{1-\gamma^2} = \frac{\gamma^4}{1-\gamma^4}.$$

Diagonalizing $AA^t$ gives $\|AA^t h\| \le fe$, since $t^2/(1+t)$ increases on $[0, \gamma^2]$. Let $\widehat{x}_m$ be the exact edge product from the actual post-start vector $s\xi$. Expanding (5) along a path of $m$ edges and applying Cauchy–Schwarz gives

$$\|x_m - \widehat{x}_m\| \le f \sum_{j=1}^{m} \gamma^{m-j} e_j \le \sqrt{S} \left( \sum_{j=1}^{m} e_j^2 \right)^{1/2}. \tag{6}$$

Hence a completed round ending on ray $L = L_w$, with terminal register $x$, acceptance $a$ and total rejection $b$, obeys

$$a + b \ge \|x\|^2 + \frac{\operatorname{dist}(x, L)^2}{S} = x^t D_L x, \qquad D_L = I + \frac{I - P_L}{S}. \tag{7}$$

Any start or erase rejection only increases the left side. Cauchy–Schwarz in the $D_L$ inner product bounds the accepted fraction by

$$\frac{|u^t x|^2}{x^t D_L x} \le u^t D_L^{-1} u = \|P_L u\|^2 + \gamma^4 (1 - \|P_L u\|^2).$$

Every round thus satisfies $a \le V(a+b)$ for the right side of (4). Summing over all completed rounds of a finite word proves its acceptance is at most $V$.

For sharpness fix a path and append $l \ge 1$ padding edges. Define

$$S_l = f^2 \sum_{j=0}^{l-1} \gamma^{2j}, \qquad D_l = I + (I - P_L)/S_l.$$

For padding write $\eta_i = \sqrt{1+\gamma^2}\, h_i$. A terminal displacement $z$ is produced by

$$\eta_i = \frac{f\gamma^{l-i}}{S_l} z, \qquad f \sum_i \gamma^{l-i} \eta_i = z, \qquad \sum_i \|\eta_i\|^2 = \|z\|^2 / S_l.$$

Take $x = D_l^{-1}u$ and $z = x - P_L x$. Choose the initial scalar so that the exact padded trajectory ends at $P_L x$; this is possible on a nonzero ray, and when $L = 0$ take scalar zero. On the original path write honestly, then use these padding displacements. Erase the old register into its reservoir with zero erase loss.

These finitely many planned vectors can all be multiplied by one sufficiently small $\tau > 0$. Choose it so that planned rejection before read is below $1/4$, every protected register plus requested scratch has squared norm below $1/2$, and the initial register has squared norm below $1/2$. The remaining total norm before read is above $3/4$. At each write there is consequently enough norm in reservoir plus scratch to produce the requested scratch while protecting the register; at each erase rotate its allowed coordinates onto their reservoir axis. Induction realizes the entire finite ideal plan. Scaling preserves its absorption ratio and leaves positive total absorption $h_0$. Its ratio is

$$\|P_L u\|^2 + \frac{S_l}{1 + S_l}(1 - \|P_L u\|^2).$$

Read leaves exactly the reset reservoir ray. Repeating this plan $N$ times gives this ratio multiplied by $1 - (1 - h_0)^N$. For each fixed $N$ approximate all its ideal rotations by actual finite-letter words. First let $N$ increase, then $l$ increase, and finally take the supremum over paths. Since $S_l \to S$, this proves (4), including zero rays. The binary conversion and size bound follow from Lemma 11.1. □

**Lemma 11.3** (general accepting effect)**.** *In the preceding proof replace terminal acceptance by $x^t F x$, where $0 \le F \le I$, retaining total terminal absorption $\|x\|^2$ and the same reset ray. For a fixed path ray $L$ every round ratio is at most*

$$\lambda_{\max}(D_L^{-1/2} F D_L^{-1/2}).$$

*Arbitrary terminal padding makes this bound sharp in the finite-word supremum.*

*Proof.* The paid-error estimate (7) is independent of the read. In matrix form, put $z = (s, h_1, \dots, h_m)$ and write the terminal register as $x = \Phi z$ by (5), with initial vector $s\xi$. Its terminal norm plus check costs is $z^t G z$, where

$$G = \Phi^t \Phi + \operatorname{diag}(0, I + A_1 A_1^t, \dots, I + A_m A_m^t).$$

The cost estimate states $\Phi^t D_L \Phi \preceq G$. Since $\Phi$ vanishes on $\ker G$, restrict its domain to $(\ker G)^\perp$ and let $G^{-1}$ denote the inverse of $G$ on that subspace. There the cost inequality is equivalent to

$$\Phi G^{-1} \Phi^t \preceq D_L^{-1},$$

since $D_L^{1/2} \Phi G^{-1/2}$ is a contraction. This is the effect-independent matrix inequality; equivalently maximize $x^t F x / x^t D_L x$ to obtain the stated eigenvalue bound. For the lower bound replace $D_L$ by $D_l$, choose a maximizing nonzero $x$, and use exactly the minimum-energy padding, scaling, erasure, repetition and finite-rotation approximation from the preceding proof. The maximum eigenvalues converge as $S_l \to S$. This also proves the assertion when $L = 0$. A realization of the read as a rational projection on a larger space, when needed, is supplied explicitly below; no rational Kraus factorization of an arbitrary effect is being assumed. □

### 11.2 A finite graph which checks an emitting computation

We specify the finite transduction and the homogeneous linear updates. Fix a deterministic tape machine $M$ which, at designated emit times, has a nonempty least-significant-first dyadic bit field on a track, bounded by delimiters and blank outside it. Multiple fields are allowed. Its finite alphabet includes all tracks and a marked-head symbol $[q, a]$ for each state $q$ and cell symbol $a$. A row is a nonempty finite cell string with exactly one marked head. Every successor row has one additional exterior blank cell on each side.

The emitter convention imposes no computational restriction. On a work track simulate a program. When it outputs a finite dyadic string, copy that string to a separate track, erase the previous field and place its two delimiters, then enter the designated emit state. A scan between stored delimiters performs copying and erasure; returning to those delimiters and extending them by one cell uses only finitely many tape states. Leave the emit state on the next step and resume simulation. A single fixed universal simulator carries the program as initial tape data. We use a universal Turing machine that simulates any machine step by step [31, 23]; Lemma 3.3 uses the same fact in its halting form. It does not require a separate transition table for each target. Only the finite initial row changes. No explicit small universal transition table or numerical dimension is claimed.

Here is a complete local successor rule. For three adjacent cells $(a, b, c)$, use the tape transition of any marked center to replace its tape symbol by the symbol written; otherwise keep the center symbol. Put the next head and state on the center if its old head stays there, the left head moves right, or the right head moves left. If none of these occurs the center is unmarked. The exactly-one-head scan makes these alternatives disjoint; reject an undefined marked transition. Blank exterior cells have no head. To transduce an old row of length $L$, start with two virtual blank cells, read old cells left to right, and emit the updated center whenever its right neighbor has arrived. After the last old cell feed two further virtual blanks. The emitted centers are precisely positions $-1, 0, \dots, L$, the desired expanded successor. The finite control stores the last two cells, the head count $0, 1$, error, and whether the row is nonempty. On the separator require count one, flush the two blanks and emit a separator. Thus the transducer has a finite explicitly specified state set and transition rule. A finite automaton also checks the emit-state and output-field format of a final row.

Assign every cell and the separator # distinct digits in $\{1, \dots, g-2\}$. Positive digits give injective base-$g$ codes of all finite strings, including their lengths. Use homogeneous registers $(c, U, V)$, initially

$$\xi = (1, 0, \mathrm{code}(C_0\#)).$$

While scanning each reported ordinary row, append its actual digits to $U$ and its transduced successor digits to $V$. Appending digit $h$ means $X \leftarrow gX + hc$; a bounded output string is the corresponding composition. The transducer state is a graph vertex, and every legal transition is one rational linear edge. After reported rows $R_0, \dots, R_{j-1}$, $U$ codes their concatenation with separators, and $V$ codes $C_0\#$ followed by their reported successors. This is an identity even when the rows are false.

A final-header edge enters a mode allowing a unary length guess $L$, each unary edge replacing $V$ by $V/g$. Close by mapping $(c, U, V)$ to $(c, E, z) = (c, V - U, 0)$. Scan the final row $W$ including its separator, requiring exactly one emit head and the regular output format. For each digit $h$ update $E \leftarrow gE - hc$, and for each output bit $d$ update $z \leftarrow (z + dc)/2$. These are again linear homogeneous updates with finite control. If $m = |W|$ and $V_0$ is the integer before the divisions, the actual final discrepancy is

$$E = g^{m-L} V_0 - \mathrm{code}(U_{\mathrm{string}} W). \tag{8}$$

The unscaled homogeneous coordinate remains $c = 1$.

**Lemma 11.4** (discrepancy gap)**.** *The final discrepancy is zero exactly for the genuine history from $C_0$ with the correct final-row length guess. Every other eligible history has $|E| \geq 1/g$.*

*Proof.* If $m > L$, the first term in (8) is an integer divisible by $g$, and the second has a positive final digit less than $g$, so their difference is a nonzero integer. If $m < L$ put $k = L - m$. When $k$ does not exceed the digit length of $V_0$, the fractional part of $V_0/g^k$ is at least $1/g$ and at most $(g-2)/(g-1)$. Its distance from any integer is therefore at least $1/g$. For larger $k$, the quotient is positive and below $1/g$, while the subtracted integer is at least one, giving the same bound. If $m = L$, the discrepancy is an integer. Its vanishing equates the complete two strings. Distinct separators then imply $R_0 = C_0$ and equality of every later row to the preceding row's transduced successor, including the final row. This includes $j = 0$, when just $W = C_0\#$ is compared. Conversely a genuine history with $L = m$ makes the strings identical. □

For an integer $D \geq 1$ a final linear edge delivers

$$(z, c - z, gDE).$$

Supply also a zero final edge, so the terminal is always reachable, and only the scalar padding loop leaves it. Scale each raw edge $B$ by the positive rational $\gamma/(1 + \sum_{ij} |B_{ij}|)$. Its norm is then at most $\gamma$ and its direction is unchanged. All updates and the graph are fixed in size. For an honest output $z \in [0, 1)$ the squared overlap against $u = e_1$ is

$$H(z) = \frac{z^2}{z^2 + (1-z)^2}.$$

For a false eligible report, $0 \leq z < 1$ and $|gDE| \geq D$, so the overlap is at most $1/(1 + D^2)$. The graph has no tests on the numerical values of its registers.

For $r$ output fields use accumulators $z_1, \ldots, z_r$, with finite control identifying the current field. Each eligible report still has $0 \leq z_j < 1$. Final rays of the form $(z_1, \ldots, z_r, gDE, 0)$ have false-history overlap at most $r/(r + D^2)$ against any unit analyzer supported in the first $r$ coordinates: Cauchy–Schwarz bounds its numerator by $\sum z_j^2 \leq r$. This variant uses $r + 2$ logical coordinates, padding the ordinary $(c, U, V)$ phase with zeros.

**A concrete emitting instance.** Take tape alphabet $\{B, 0, 1\}$, states $P, Q, E$, initial row $[P, B]$, and transitions

$$\delta(P, B) = (Q, 1, +1), \quad \delta(Q, B) = (E, 0, 0), \quad \delta(E, 0) = (P, 0, +1).$$

State $E$ emits. Here the output field is the block of 0's and 1's in the row (format $B^*[01]^+B^*$), without delimiters. Rows expand as above, and emit rows at times $2, 5, 8, \ldots$ carry the LSB-first strings $10, 1010, 101010, \ldots$, hence

$$z_k = \frac{1 - 4^{-k}}{3} \longrightarrow \frac{1}{3}.$$

Number the three unmarked symbols by $1, 2, 3$, and the nine marked symbols by $4, \ldots, 12$, with $[P, B] = 4$, $[Q, B] = 7$, $[E, B] = 10$; let $\# = 13$ and $g = 15$. Then $\mathrm{code}(C_0\#) = 73$. The preceding transition rule and digit-update rules specify the graph without a hidden interpreter oracle. With $D = 4$, $\gamma = 1/2$ the honest overlap supremum is $H(1/3) = 1/5 > 1/17$, so Theorem 11.2 gives value $1/16 + (15/16)(1/5) = 1/4$. The first three path values are $5/32$, $521/2336$ and $1781/7328$. The accompanying exact check uses this example; the general proof is Lemma 11.4 and does not depend on it.

### 11.3 Every left-c.e. real, with fixed dimension

**Theorem 11.5** (one-way KW spectrum)**.** *The values of rational one-way KW automata are exactly the left-c.e. reals in* $[0,1]$*. Two real rational orthogonal letter gates, three fixed coordinate outcome projections, a basis start, a rejecting right marker and one fixed dimension suffice. One continuing block is a finite-order permutation and the other a partial isometry. The fixed dimension is not computed and is large.*

*For* $0 < \alpha \leq 1$ *a realizing description is computable from an ordinary lower-approximation program and a rational* $0 < l < \alpha$*. Equivalently, an index alone suffices under the promise* $\alpha > 0$*. No total compiler valid on all of* $[0,1]$ *exists. These conclusions and the upper inclusion also hold when algebraic data are allowed.*

*Proof.* Given $l$, choose rational $\gamma$ and integer $D$ with

$$\gamma^4 + (1-\gamma^4)/(1+D^2) < l.$$

The function $H$ increases continuously from zero to one on $[0,1]$. Dovetail over dyadics $t \in [0,1)$ and witnesses to

$$\gamma^4 + (1-\gamma^4)H(t) < \alpha.$$

Output the running maximum, initially zero. These maxima converge to the unique $y$ with $\gamma^4 + (1-\gamma^4)H(y) = \alpha$, including $y = 1$. A fixed emitting interpreter performs this computation from its loaded program. Its false-history ceiling is below $H(y)$, so its graph overlap supremum is exactly $H(y)$. Theorem 11.2 gives value $\alpha$. The graph and register format are independent of the program, $l$, $\gamma$ and $D$; only rational coefficients and the initial direction vary. Binary conversion has a fixed size. Always rejecting gives zero in the same padded size.

For the upper inclusion enumerate all finite words and their exact rational probabilities, taking running maxima. With algebraic or computable data instead dovetail certified rational lower estimates to each finite-word probability. This uniformly gives a left-c.e. value. A positive-name promise lets us wait for a positive approximant and take half of it as $l$.

To prove the last impossibility, positivity is decidable for rational KW descriptions. Starting with $\psi$, compute the span of reachable vectors $T_w\psi$, closing under all letters until its dimension stops growing. Finite-dimensional rational linear algebra makes this effective. There is positive ordinary or marker acceptance exactly when one of their accepting amplitude maps is nonzero on that span; a map vanishes on all reachable vectors exactly when it vanishes on their span. Include any immediate left-marker acceptance. This also works with exact algebraic arithmetic. A total compiler for the lower name zero until a machine halts and $1/2$ thereafter would therefore decide halting. In particular the released noncomputable $\alpha_*$ in Lemma 6.1, compiled here with $l = 1/8$, is a single noncomputable rational KW value. Both parts of Question 9.4 are false. □

The compiler from a lower name and a positive lower bound concerns KW automata. It does not answer the question for probabilistic automata after Proposition 8.1, since Corollary 11.6 transfers values only from probabilistic to KW automata.

The existence bound is not a small-device claim. An independent instantiation of the three-state, three-symbol example above already has 2,844 live coordinates and 2,858 logical letters before binary conversion, giving total binary dimension 24,384,456. A universal emitter is larger; we use the fixed-but-not-computed convention of Theorem B. These reported counts are not a dimension calculation for the universal construction.

**Corollary 11.6** (a useful PFA transfer)**.** *For a rational PFA of value $v$, one can effectively construct a rational KW automaton of value $\gamma^4 + (1-\gamma^4)v^2/(1+v^2)$.*

*Proof.* Use column-stochastic matrices and scale each transition positively to norm at most $\gamma$. A final edge sends a stochastic column $x$ to $(bx, \mathbf{1}^t x, 0, \ldots)$, with $b$ its accepting row, and is likewise scaled. The resulting ray is $(z, 1, 0, \ldots)$ for PFA acceptance $z$. Allow all data words, this final edge and then only padding. The overlap is $z^2/(1+z^2)$, so Theorem 11.2 and continuity prove the formula. This is a transfer of optimized values, not of individual word probabilities. □

# 12 Full spectra with fixed hardware

Here "fixed" includes the letter matrices, accepting set or measurement, dimension and right-marker rule. Only a rational initial distribution or rational real unit vector varies. In KW's original basis-start convention the latter means varying the left-endmarker gate on that basis start: a rational Householder reflection prepares any rational unit vector. The ordinary gates, measurement and right marker remain fixed.

## 12.1 Addresses and bins

Enumerate all programs printing rationals, ignoring malformed output, clamping to $[0,1]$ and taking running maxima from zero. Write $\alpha_e$ for their limits; every left-c.e. real in $[0,1]$ occurs, and bounded-stage approximants are computable uniformly in $e$. Put

$$n = e + 4, \quad w_e = 2^{-n}, \quad h_e = 2^{-3n}, \quad \delta_e = h_e^2, \quad K_e = 1/\delta_e.$$

Place $K_e$ addresses $y_{ek}$ on a $2^{2n} \times 2^{2n} \times 2^{2n}$ grid of spacing $h_e$ in the cube beginning at $(4w_e, 0, 0)$, in lexicographic order. Its side is less than $w_e$. For $i = (e,k)$ set

$$b_i = \max(0, \min(\delta_e, \alpha_e - k\delta_e)).$$

There are uniform dyadic increasing approximants $b_{is} \uparrow b_i$, obtained by rounding lower approximants down on successively finer dyadic grids and taking maxima.

**Lemma 12.1** (address separation)**.** *For different addresses $i, j$,*

$$\|y_i - y_j\|^2 \geq \max(\delta_i, \delta_j), \qquad \|y_i\|^2 < 27/256.$$

*The rational nonnegative unit vectors*

$$u_i = \frac{(2y_i, 1 - \|y_i\|^2)}{1 + \|y_i\|^2} \tag{9}$$

*satisfy $1 - (u_i^t u_j)^2 \geq \max(\delta_i, \delta_j)/2$.*

*Proof.* Inside a cube the separation is at least its grid spacing. A smaller cube has first coordinate below $(5/2)w_e$, whereas cube $e$ begins at $4w_e$, giving more than the required separation across cubes. The coordinate bounds are $y_1 < 5/16$ and $y_2, y_3 < 1/16$. Expansion gives

$$\|u_i - u_j\|^2 = \frac{4\|y_i - y_j\|^2}{(1+\|y_i\|^2)(1+\|y_j\|^2)} \geq \|y_i - y_j\|^2.$$

All coordinates are nonnegative. Thus $c = u_i^t u_j \geq 0$ and $1 - c^2 \geq 1 - c = \|u_i - u_j\|^2/2$. □

## 12.2 One fixed probabilistic automaton

**Theorem 12.2** (fixed PFA)**.** *There are two fixed rational stochastic matrices and a fixed accepting set whose values, as the rational initial distribution varies, are exactly all left-c.e. reals in* $[0,1]$.

*Proof.* The lower semicomputable convex function

$$F(y) = \sup_{i,s} (2y_i \cdot y - \|y_i\|^2 + b_{is})$$

obeys $F(y_j) = \|y_j\|^2 + b_j$. Indeed a rival term equals $\|y_j\|^2 - \|y_i - y_j\|^2 + b_{is} \le \|y_j\|^2$, by Lemma 12.1, and the own terms approach the asserted value.

Fix $a = 1/2$, $B = 1/64$. Uniformly enumerate the dyadic five-vectors

$$q_0 = a + B(-\|y_i\|^2 + b_{is}), \quad q_j = q_0 + 16B(y_i)_j \ (1 \le j \le 3), \quad q_4 = q_0 + 8B.$$

All entries belong to $[1/4, 3/4]$. Encode their equal-length binary fractional expansions by interleaving one bit of each coordinate cyclically, in least-significant-first order. Padding the least-significant end with zeros equalizes lengths. This is one fixed c.e. language, so Lemma 3.2 gives one fixed counter program and Lemma 4.5 and Corollary 4.6 give its fixed certificate checker. Use verifier base $B_0 = 32$, so $\kappa = (32 + 1/32)/2$.

Make five permanent disjoint types. Each runs the same parser and tests; its independent binary register updates only on its own output bits, with the matrices $R_0, R_1$ of Section 4. On a well-formed closing delimiter absorb all live joint and split mass: joint register one accepts, joint register zero rejects, and all split mass rejects. Reset only dead mass to that type's fixed fresh parser distribution. Syntax errors reject. On any certificate the joint and split masses $\Phi, \Psi$ are identical across types, so total absorption $\Phi + \Psi$ is type independent, including on false certificates. Consequently each reset preserves the original relative type prior $p$.

For a valid certificate of output vector $q$, the absorption fraction is $2\Phi > 0$, and type $j$ pays fraction $q_j/2$. On a false certificate its payoff is at most $1/(1+\kappa) < 1/8$, whereas every honest payoff is at least $1/8$. Registers on false outputs still lie in $[0,1]$. Malformed and incomplete portions add no acceptance. Summing round inequalities, as in Lemma 5.2, and repeating a valid certificate, as in Lemma 5.3, proves

$$V(p) = \frac{1}{2} \sup_{i,s} \sum_{j=0}^{4} p_j q_j(i,s). \tag{10}$$

No reset row contains $p$: each type has its own fixed reset distribution. This is the change required to fix the hardware rather than only its size.

For rational $y$ at an address and $t \in [0,2]$ set $p_j = y_j/8$ for $j = 1,2,3$, $p_4 = t/8$, and let $p_0$ be the remainder. These are probabilities by the coordinate bounds. Substitution in (10) gives

$$V(p) = [a + B(F(y) + t)]/2.$$

Choose a bin $k$ with $k\delta_e \le \alpha_e \le (k+1)\delta_e$, using the first or last bin at endpoints. At its address take $t = 1 + k\delta_e - \|y_{ek}\|^2 \in [0,2]$. The value becomes $[a + B(1 + \alpha_e)]/2$. Hence all left-c.e. targets in the calibrated interval

$$I = [33/128, 17/64]$$

occur with fixed matrices. Given any $\alpha \in (0,1)$ choose rational $d > 0$ small and rational $c \ge 0$ with $c + d \le 1$ and $(\alpha - c)/d \in \operatorname{int} I$; density supplies $c$ after taking $d < \min(\alpha, 1-\alpha)/2$.

Affine normalization is left-c.e. and therefore has an index in our enumeration. Put mass $d$ on its calibrated start, mass $c$ on an added accepting sink and the remaining mass on a rejecting sink. The value is $\alpha$. The sinks alone realize endpoints. All these sinks and transitions are fixed. Lemma 7.1 makes the alphabet binary without changing its matrices as the prior varies. Every rational prior conversely has a left-c.e. value by Proposition 2.3. □

Rote's fixed-matrix emptiness theorem [28] concerns undecidability of a strict threshold with the initial distribution as input. The preceding theorem concerns the entire set of numerical suprema, and does not follow just from that undecidability statement.

### 12.3 One fixed quantum apparatus

**Lemma 12.3** (rational four squares)**.** *Every nonnegative rational is the squared norm of a rational vector of length four, and such a vector can be found effectively.*

*Proof.* The quaternion norm $N(x) = \sum_{j=1}^{4} x_j^2$ is multiplicative, by $x\bar{x} = N(x)$ and reversal of products under conjugation. It suffices to represent primes. The prime two is immediate. For an odd prime $p$, the two residue sets $\{x^2 : 0 \le x \le (p-1)/2\}$ and $\{-1-y^2 : 0 \le y \le (p-1)/2\}$ each have $(p+1)/2$ elements and intersect. Thus $mp = x^2 + y^2 + 1$ for some $1 \le m < p$.

Minimize the positive multiplier $m$ among all four-square representations $mp = N(X)$. If $m$ is even, pair equal-parity coordinates and replace each pair $(s,t)$ by $((s+t)/2, (s-t)/2)$. There are an even number of odd coordinates, so this gives integer coordinates of norm $mp/2$, a contradiction. For odd $m > 1$, choose centered residues $R$ of $X$ modulo $m$. Then $N(R) = mm'$ with $0 \le m' < m$, since $|R_j| \le (m-1)/2$. The case $R = 0$ would imply $m \mid p$, impossible for $1 < m < p$. The quaternion $X\bar{R}$ is coordinatewise divisible by $m$, because modulo $m$ it is the scalar $N(R)$. Its quotient has norm $pm'$, contradicting minimality. Hence $m = 1$. Multiply prime representations to obtain any nonnegative integer. To represent $u/v$, represent the integer $uv$ and divide the four coordinates by $v$. Exhaustive finite integer search finds the representation, proving effectivity. □

**Theorem 12.4** (fixed rational pure quantum starts)**.** *One fixed rational KW apparatus with two real rational orthogonal letters, fixed coordinate measurement and rejecting right marker has exactly the full left-c.e. spectrum over its rational real unit initial live vectors. Its continuing blocks have the permutation and partial-isometry forms of Lemma 11.1.*

*Proof.* Tensor each live phase of Theorem 11.2 with a $d$-dimensional reference. Before read all continuing operations act as $I \otimes T$. At read accept the rational Bell-line projector $\Omega\Omega^t/d$ on reference tensored with logical register, where $\Omega = \sum_j e_j \otimes e_j$; reject its orthogonal complement there, and continue reference tensored with reservoir to reset. On $u \otimes x$ acceptance is $|u^t x|^2/d$, while terminal absorption is $\|x\|^2$. On continuation the same reference factor $u$ remains. Therefore for each word acceptance is precisely the analyzer-$u$ probability divided by $d$, with unchanged live evolution. Theorem 11.2 gives work value

$$W(u) = \left[\gamma^4 + (1-\gamma^4) \sup_w \|P_{L_w} u\|^2\right]/d. \tag{11}$$

The rational projector is its own Gram factor; no gate entry $1/\sqrt{d}$ is required. Tensoring and rational completion preserve the required model.

Use the addresses of Lemma 12.1, padded with zeros, and put $p_0 = 9/25$, $\kappa = 1/16$. A single fixed program dovetails over all $i, s$ and rational nonnegative circle points $(a, b)$, emitting

$$v = (au_i, b), \qquad a^2 + b^2 = 1,$$

when $p_0 \le a^2 < p_0 + \kappa b_i$ is witnessed by a lower approximant; also emit $(a, b) = (3/5, 4/5)$ at every address. Rational circle points are dense, by the parametrization $((1 - t^2)/(1 + t^2), 2t/(1 + t^2))$. The own-address overlap supremum is $p_0 + \kappa b_i$. For $i \neq j$, the overlap of a ray from address $i$ against analyzer $u_j$ is at most

$$(p_0 + \kappa\delta_i)(1 - \delta_i/2) \le p_0,$$

since $\kappa \le p_0/2$.

Convert each emitted rational vector to five dyadic fields by multiplying by a common integer denominator and dividing all its integer coordinates by one sufficiently large power of two. This preserves its nonzero ray and makes each field less than one. Use the multi-field verifier of Section 11.2, with $d = 7$ and $D = 8$, final ray $(z_1, \dots, z_5, gDE, 0)$. False-history overlaps are at most $5/69 < p_0$. Thus the complete projective supremum on analyzer $u_i$ is exactly $p_0 + \kappa b_i$. Fix $\gamma = 1/2$. The entire graph, its initial logical direction and its letter maps are now fixed.

Add a one-time initialization phase with work space and separate four-coordinate yes and no sectors. Its launch maps work isometrically to reference tensored with reset reservoir, accepts the yes sector and rejects the no sector; every other operation rejects this initial phase. Work never returns there. A rational preparation with work amplitude $h$, yes mass $c$ and no mass $1 - h^2 - c$ consequently has value $c + h^2 W(u)$. Lemma 12.3 supplies rational amplitudes for these masses.

Take $h = 1/2$ and define

$$C = \frac{\gamma^4 + (1 - \gamma^4)p_0}{4d} = \frac{1}{70}, \qquad \eta = \frac{\kappa(1 - \gamma^4)}{4d} = \frac{15}{7168}.$$

At address $(e, k)$ put $c = \eta k\delta_e$. The value is $C + \eta(k\delta_e + b_i)$, hence $C + \eta\alpha_e$ in a correct bin. All masses are nonnegative and sum to one. This realizes every left-c.e. real in $[C, C + \eta]$.

Add one further fixed outer initialization of the same kind. For $\alpha \in (0, 1)$ choose a small positive rational square $h^2$ and rational $c \ge 0$ such that $c + h^2 \le 1$ and $(\alpha - c)/h^2 \in (C, C + \eta)$. Rational density makes this possible. Give work the amplitude $h$ times the entire inner preparation, and choose four-square witnesses for the yes and no masses. The value is $\alpha$; pure yes or no realizes the endpoints. Every operation again has projection effects and a partial-isometry continuing block. Binary conversion completes the construction; the start occupies copy zero. Conversely word enumeration gives a left-c.e. value for every rational start. □

No closed-interval selector degree for this particular Bell-read hardware is asserted. The proofs below use its explicitly established interior calibration only.

## 13 Isolation, density and arithmetic index sets

All completeness assertions use computable many-one reductions on finite descriptions. A $\Sigma^0_k$ predicate has $k$ alternating number-quantifier blocks beginning existentially and a decidable matrix; $\Pi^0_k$ denotes its complement class. Algebraic numbers are represented effectively by

polynomials and isolators. Strict comparisons between a computable real and a rational are semidecidable, by certified approximations with a strict margin.

**Lemma 13.1** (interval closure)**.** *Uniformly from any machine constructed in Theorem 11.5, a rational binary machine of fixed size can be built for which the closure of the set of finite-word probabilities is exactly* $[0, V]$*. It retains the basis start, rejecting marker and continuing-block restrictions.*

*Proof.* Add a one-time two-dimensional prephase with basis start. Its rotation letter is the $3/5, 4/5$ rotation; its launch maps its first coordinate to the old start and rejects the second. Other old letters reject this phase; prephase operations reject the old space. The old reset never returns to the prephase. Every probability is zero or $\cos^2(n\theta)P(w)$, where the cosine-square factors are dense in $[0, 1]$. No probability exceeds $V$. For $s < V$, take an old word of probability greater than $s$ and approximate the requisite factor; approach $V$ with unattenuated old words. This gives the entire closure, also at $V = 0$. Rational completion and binary conversion preserve word probabilities and therefore the closure. □

**Theorem 13.2** (value one and isolation)**.** *For rational one-way KW automata the following yes-index sets are* $\Pi^0_2$*-complete: value one; nonisolation of a specified cutpoint, already at the fixed cutpoint* $1/2$*; and density of the finite-word probabilities in* $[0, 1]$*. Isolation of a specified cutpoint, and existence of an isolated real cutpoint in* $[0, 1]$*, are* $\Sigma^0_2$*-complete. Hardness holds for real rational orthogonal binary automata of one fixed size, with basis start and rejecting right marker. The upper bounds, and hence the completeness statements, hold also for algebraic descriptions and algebraic cutpoints. For the specified-cutpoint upper bound a computable cutpoint suffices.*

*Proof.* Writing $f(w)$ for finite-word acceptance, the three upper bounds are

$$\begin{aligned} V = 1 &\iff \forall n\, \exists w \quad f(w) > 1 - 2^{-n}, \\ \lambda \text{ nonisolated} &\iff \forall n\, \exists w \quad |f(w) - \lambda| < 2^{-n}, \\ \{f(w)\} \text{ dense} &\iff \forall I\, \exists w \quad f(w) \in I, \end{aligned}$$

where $I$ ranges over rational open subintervals of $(0, 1)$. Exact algebraic comparison, or a finite witness to a strict computable-real comparison, gives the claimed quantifier form. An isolated real cutpoint exists precisely when the closure of the probability set is not $[0, 1]$. Indeed a nonempty relative open complement contains an interior rational interval. This also shows that allowing cutpoints in $(0, 1)$ instead gives the identical existence question.

For hardness take any decidable relation $R(e, j, k)$ and put

$$m_s = \max\{m \le s : (\forall j < m)(\exists k \le s) R(e, j, k)\}.$$

This is nondecreasing and unbounded exactly when $\forall j \exists k\, R(e, j, k)$. Compile the lower name $1 - 2^{-m_s - 1}$ with supplied bound $1/4$, and apply Lemma 13.1. Its value equals one, and its probability set is dense in $[0, 1]$, exactly in the unbounded case. For nonisolation at $1/2$ use instead the name $1/2 - 2^{-m_s - 2}$ and bound $1/8$. Its probability closure reaches $1/2$ exactly in that case. These are uniform computable maps of indices and prove all three hardness assertions. Taking complements proves the two isolation classifications. □

This settles Problems (6) and (7) of Derksen, Jeandel and Koiran [9, p. 359] for the Kondacs–Watrous measure-many model: isolation of a given threshold, and existence of an isolated threshold. They record there that Problems (1)–(5) were known to be undecidable for this model, by work of Jeandel from 2002, and that the status of (6) and (7) was unknown. Their threshold convention is $[0, 1]$; the later survey [1, Section 4.5.2] uses $(0, 1)$ for the specified

threshold. Hardness at $1/2$ covers both conventions, and the closure argument covers both existence conventions. The conclusion extends to NaQFA and BiQFA, which contain KW acceptance functions, but leaves Latvian values and isolation open.

These are quantum counterparts of classical results, not new PFA isolation theorems: undecidability of PFA cutpoint isolation goes back to Bertoni [33] and Bertoni, Mauri and Torelli [34], and Chadha, Sistla and Viswanathan proved its $\Sigma_2^0$-completeness [36]; PFA value-one undecidability is due to Gimbert and Oualhadj [15], with the subsequent arithmetic discussion in [37]. We use the explicit $\Pi_2^0$ yes-set formula above; Turing equivalence between a decision problem and a complement does not preserve many-one $\Sigma/\Pi$ polarity. Existing PFA-to-KW strict-cutpoint simulations preserve the sign relative to a threshold, with length-dependent attenuation [1, 44]. Such attenuation need not preserve a gap or a supremum. Our reduction needs the quantitative projection theorem and is not an immediate transfer of classical isolation.

## 13.1 Arithmetic of the optimum

**Theorem 13.3.** *For rational KW descriptions, rationality and algebraicity of the value are $\Sigma_3^0$-complete; transcendence and the property of being a Liouville number are $\Pi_3^0$-complete. Hardness already holds in the fixed-size binary class of Theorem 11.5. Here a Liouville number is irrational and for every $j \geq 1$ has a rational $a/b$, $b \geq 2$, with $0 < |V - a/b| < b^{-j}$.*

*Proof.* Let $v_s$ be the increasing rational word maxima. Equality to any fixed computable algebraic $t$ is $\Pi_2^0$: require all maxima to be at most $t$, and arbitrarily close from below. Enumerating rationals or real algebraic numbers gives the two $\Sigma_3^0$ memberships. Algebraic enumeration and comparison are effective, for example by squarefree integer polynomials with rational isolators. Transcendence is the complement of algebraicity. For fixed $j, a, b$, $V < a/b + b^{-j}$ is $\Sigma_2^0$: some rational strictly below that endpoint bounds every $v_s$. The other strict inequality is c.e. Irrationality excludes equality with $a/b$. Quantifying over all $j$ and the witnessing rationals yields $\Pi_3^0$ for Liouville numbers.

For completeness we recall the movable-marker reduction behind the classical index set Cof [42, Chapter IV, Section 3]; see also [41, Section 14.8]. Given $\exists i \forall j \exists k\, R(e, i, j, k)$, let each label $i$ have a nondecreasing progress counter, the largest $m \leq s$ for which all $j < m$ have witnesses by stage $s$. At stage $s$ maintain one marker for each label $i \leq s$. Whenever counter $i$ increases, enumerate into a c.e. set $W$ every current marker of label $h \geq i$, cancel them, and assign missing labels distinct fresh values above the stage and all previously issued values. Enumerate every number at most the stage which is not a current marker. Every number outside $W$ is thus a permanent marker. If some counter increases infinitely, only finitely many smaller labels can have permanent markers; $W$ is cofinite. If every counter increases finitely, each label eventually has a permanent marker, so the complement is infinite. All stage operations are finite and effective.

Put $f(n) = (n+3)!$ and

$$\alpha = \frac{3}{4} - \sum_{n \notin W} 2^{-f(n)}.$$

This is uniformly left-c.e., with a known bound above $1/2$: start with increasing lower estimates for the computable constant $3/4 - \sum_n 2^{-f(n)}$ and add enumerated terms. The series tail is bounded by twice its first omitted term. A finite complement makes $\alpha$ rational. For an infinite complement, truncate after a missing index $n$. The partial rational has denominator $q = 2^{f(n)}$ and strictly positive error at most $2 \cdot 2^{-f(n+1)}$. These infinitely many distinct rationals

approach too fast for a rational limit: a distinct rational of denominator $b$ is at distance at least $1/(bq)$. They are Liouville approximations because $f(n+1)/f(n) = n+4 \to \infty$. An irrational algebraic limit of degree $k$ is impossible as well: for an integer polynomial $P$ vanishing at it, $|P(a/q)| \geq q^{-k}$ unless zero, while a bounded derivative gives $|P(a/q)| \leq C|\alpha - a/q|$; only finitely many distinct approximants can be roots. Thus the infinite-complement case is transcendental Liouville. Compile with strict lower bound $1/4$. The two alternatives simultaneously prove all four hardness assertions. □

**Theorem 13.4.** *The rational KW descriptions with a computable value form a $\Sigma^0_3$-complete set, with hardness in the same binary fixed-size class. For any two fixed rational targets $0 < a < b \leq 1$, no uniform algorithm distinguishes devices of value $a$ from those of value $b$, even under that two-value promise.*

*Proof.* The value is computable iff there is a total computable convergence modulus: on input $k$ it returns $n$ such that $v_m \leq v_n + 2^{-k}$ for every $m$. A computable value supplies a modulus by searching for a maximum close to a two-sided approximation with a strict margin; a modulus supplies two-sided approximations. Thus membership is $\exists e \forall (k, m) \exists t$: by time $t$, program $e(k)$ has returned $n$ and the stated rational inequality holds. The output is unique, so these quantifiers express exactly the required modulus.

For hardness use the progress counters in the preceding proof and the standard Rec marker diagonalization [42, 41]. On assigning a fresh marker $x$ to label $h$, simulate putative characteristic program $h$ on $x$. If it returns zero, enumerate $x$ into $W$ and leave the label finished until injury. If it returns one, reserve $x$ outside $W$ until injury. While the computation is pending, or if it returns any other output, $x$ stays reserved. Each progress increase of counter $i$ cancels all reservations and finished states of labels $h \geq i$, enumerating canceled reservations into $W$ and assigning fresh markers as before; enumerate every unreserved number below the stage. If a counter progresses infinitely, $W$ is cofinite. If every counter progresses finitely, every total characteristic program disagrees with $W$ at its label's final marker, so $W$ is not recursive.

Target

$$\alpha = \frac{1}{4} + \frac{1}{2} \sum_{n \in W} 4^{-n-1}.$$

A cofinite $W$ gives a rational value. Conversely a computable $\alpha$ decides $W$ by successive digits: a present digit contributes its full weight, while the later tail is at most one third of that weight, exactly the gap argument of Lemma 6.1. In the nonrecursive case the value is therefore noncomputable. Compile with $l = 1/8$, proving hardness. Finally compile the lower program constant at $a$ until a given computation halts and constant at $b$ thereafter, using $l = a/2$. A uniform promised separator would decide halting. □

The effective ingredients in these arguments are the uniform lower enumeration of a device's value and the compiler from positive lower names with a supplied strict positive bound. We apply them to the specific bounded-away-from-zero names above; no general assertion that an arbitrary property survives removal of the zero target is needed.

## 14 The degrees of exact preparation selectors

A selector for a specified fixed apparatus takes an *ordinary program index* of a lower name of $\alpha$ in the promised target set, halts, and returns a finite rational preparation of exactly that value.

Its oracle is fixed independently of the input. The degree $0'$ is the halting degree, and $0''$ its jump. "Least" means an upper construction with that oracle and a Turing reduction of that oracle to every oracle which computes a selector. Finite facts about a specified fixed apparatus may be hard-coded; such assertions need not be uniform in the apparatus.

**Lemma 14.1** (fixed-precision obstruction)**.** *For every fixed finite PFA or KW apparatus, any exact selector on a nondegenerate interior interval computes* $0'$*. In particular no computable selector exists there.*

*Proof.* For PFA starts $p, p'$, every word payoff lies in $[0,1]^d$, so $|V(p) - V(p')| \le \|p - p'\|_1$. For unit quantum starts, write word acceptance $\psi^* E_w \psi$ with $0 \le E_w \le I$; subtracting forms gives $|V(\psi) - V(\phi)| \le 2\|\psi - \phi\|$. On the compact simplex or sphere, for any one fixed error tolerance there is a finite rational net and a finite table of rational approximations to the values at its centers. Such a table exists even if no uniform algorithm computes it as the tolerance varies. Rational unit centers have effective finite nets by stereographic parametrization and the omitted pole, or by enumerating rational centers and testing finite coverage. Hard-code a table fine enough to distinguish two fixed separated rationals in the promised interval. Feed a selector the lower program staying at the smaller rational until a proposed machine halts, then staying at the larger. Applying the fixed table to its returned preparation decides halting. This proves the Turing lower bound, not just the failure of a particular computable inverse. □

**Theorem 14.2** (interior and classical endpoint degrees)**.** *For the fixed PFA of Theorem 12.2 and the fixed pure quantum apparatus of Theorem 12.4, the least selector degree on* $(0,1)$*, or on any promised nondegenerate interval within it, is* $0'$*. For every fixed full-spectrum PFA the least selector degree on* $[0,1]$ *is* $0''$*. An upper bound* $0''$ *holds for every specified fixed computable apparatus with enumerable rational preparations and full left-c.e. spectrum. The latter general statement is only an upper bound for quantum apparatuses.*

*Proof.* Oracle $0'$ decides each strict query $q < \alpha$, since it decides whether the search for a lower approximant above $q$ halts. Bisection therefore gives two-sided approximations to $\alpha$. For an interior target, search for rational outer coefficients placing it strictly inside the calibrated interval $I$ in the PFA proof, or $[C, C + \eta]$ in the pure quantum proof. Strict placement is recognizable from the approximations; in the latter search take the work coefficient to be a positive rational square. The existence arguments in Theorems 12.2 and 12.4 guarantee termination.

Once these rational coefficients are fixed, affine normalization of the original ordinary lower program is another ordinary program. Compute its index $e$ in the fixed clamped enumeration by the parameter theorem (the *s-m-n* step); the new program contains no oracle. Use $0'$ for the finitely many strict tests $k\delta_e < \alpha_e$, taking the largest allowed $k$ with $0 \le k < K_e$, defaulting to zero. At an internal grid equality this picks the preceding bin; its upper endpoint is allowed. At target one it picks the last bin. All remaining address arithmetic, type probabilities and four-square choices are effective. The calibrated start and outer launch or sink mixture give the required exact preparation. Lemma 14.1 gives optimality for interior targets.

For the endpoint lower bound fix any PFA. Whether $V(p) = 1$ depends only on $S = \operatorname{supp} p$. If word probabilities tend to one from $p$, each basis-state deficit on $S$ tends to zero, since its fixed positive weight times its nonnegative deficit is bounded by the total deficit. The same word sequence then tends to one from any prior with support $S$. There are finitely many supports, so, for this fixed PFA, a finite hard-coded table decides $V(p) = 1$ on all rational priors. A selector on $[0,1]$ followed by this table decides whether an arbitrary lower

name has supremum one. The progress-counter names in Theorem 13.2 show that this is $\Pi^0_2$-complete, equivalently it computes $0''$. This argument applies to every fixed full-spectrum PFA, regardless of the geometry of its preparation map.

For the general upper bound enumerate its rational preparations. Their values have uniform lower names. Equality of two lower-named reals is $\Pi^0_2$: every witnessed rational strict lower bound of either must eventually be witnessed for the other. Oracle $0''$ decides this condition. Search for a preparation whose value equals the input target. Full spectrum guarantees termination, proving the upper bound and the exact classical closed-interval degree. □

The additional jump for the PFA occurs at target one. The displayed constructions have degree $0'$ also on $[0, 1)$: decide positivity with $0'$, return a fixed zero start if necessary, and otherwise use the interior selector. For every fixed full-spectrum PFA the lower bound $0''$ already holds on $(a, 1]$, for rational $0 \leq a < 1$, by affine rescaling of the progress-counter targets to stay above $a$.

**Proposition 14.3** (streams are a different representation)**.** *No fixed oracle selects finite exact preparations from lower-approximation streams on a nondegenerate interval, even if all supplied streams are computable. The same holds for streams enumerating strict left cuts.*

*Proof.* Fix the proposed oracle and two rationals $a < b$ in the interval. On the constant stream $a$ a halting computation queries finitely many positions. A computable nondecreasing stream identical through all these positions and equal to $b$ thereafter receives the same finite output, which cannot have both values. For strict-cut enumerations, keep the finitely queried prefix of an enumeration of the rationals below $a$, then continue to enumerate all rationals below $b$. This proves the same contradiction under that convention. □

This is an instance of the standard finite-use, or continuity, principle for computation on names of reals [43].

Consequently the input format matters: $0'$ decides an ordinary program's future witness search from its index, but does not uniformly decide a search relative to an arbitrary stream supplied as an oracle. We make no assertion here about selectors from input-dependent jumps of arbitrary streams.

## 15 A quantum selector on the closed interval

This section specifies another apparatus, using a symmetric read instead of the Bell-line read. It proves an existence statement about this apparatus, not an endpoint degree for all fixed quantum hardware. All claims specific to this construction are confined to this section. As in Theorem 12.4, every live work phase carries a reference factor and every continuing work map before a read acts as $I \otimes T$. After successful launch, each preparation used in the construction is a product of its reference with the reset reservoir, so continuation retains that reference throughout the work evolution.

**Lemma 15.1** (symmetric read)**.** *At $\gamma = 1/2$, replace the reference-register read by the rational projector $(I + \mathrm{SWAP})/2$ onto the symmetric tensor subspace. Continue reference tensored with*

*reservoir. If at least one nonzero path ray exists, the work value from reference $u$ is $G(p)$, where*

$$p = \sup_w \|P_{L_w} u\|^2, \tag{12}$$

$$G(p) = \frac{18 + 15p + \sqrt{(18+15p)^2 - 128}}{64}. \tag{13}$$

*Here $G$ increases continuously from $1/2$ to $1$, with rational inverse*

$$J(\beta) = \frac{32\beta + 1/\beta - 18}{15}, \qquad 1/2 \le \beta \le 1. \tag{14}$$

*Proof.* On $u \otimes x$ the accepted mass is $(\|x\|^2 + |u^t x|^2)/2$, by direct expansion of the swap. Thus the logical effect is $F = (I + uu^t)/2$; total terminal absorption remains $\|x\|^2$ and continuation retains the same reference. The swap identity is the usual one behind the swap test [35]; this realization is a single rational-projector KW read, completed by Lemma 11.1.

For a nonzero path ray let $P = P_L$, $p = \|Pu\|^2$ and $D = P + 16(I - P)$. By Lemma 11.3 the optimized round ratio is the largest eigenvalue of $D^{-1/2} F D^{-1/2}$. On the plane of $L$ and $u$ its trace is $(18 + 15p)/32$ and determinant $1/32$; orthogonal directions have eigenvalue $1/32$. These identities follow by writing $u = \sqrt{p}\, l + \sqrt{1-p}\, v$ in an orthonormal plane and multiplying by $\mathrm{diag}(1, 1/4)$ on both sides of $F$. They extend to $p = 0, 1$ by continuity. The larger root of

$$32\lambda^2 - (18 + 15p)\lambda + 1 = 0$$

is (13). Padding attains its supremum by Lemma 11.3; repetition retains the reference. For a zero ray $D = 16I$ and the ceiling is only $1/16$, so zero rays cannot exceed the nonzero-ray floor $1/2$. Taking the supremum of nonzero rays proves the formula. Substitution proves (14); $\beta \ge 1/2$ selects the larger root. Moreover

$$J(b) - J(a) = (b - a)(32 - 1/(ab))/15 > 0 \quad (a < b), \tag{15}$$

$$1 - J(\beta) = (1 - \beta)(32 - 1/\beta)/15 \le (31/15)(1 - \beta). \tag{16}$$

These show monotonicity and the endpoint values. □

**Theorem 15.2** (five-coordinate closed-interval selector)**.** *There is one fixed binary real rational orthogonal KW apparatus with fixed coordinate outcomes and rejecting right marker which realizes every left-c.e. real in $[0, 1]$ by nonnegative rational pure starts supported on five fixed coordinates. Its least exact selector degree, for ordinary lower-program indices on the closed interval, is $0'$. On this same apparatus, value one on rational pure starts is $\Pi^0_2$-complete under computable many-one reductions. One continuing block is a finite-order permutation and the other a partial isometry.*

*Proof.* Use a padded enumeration $\alpha_e = \alpha_j$ for $e = \langle j, t \rangle$, with a computable pairing unbounded as $t$ grows. An ordinary lower program is converted effectively to its printed-stream index $j$. Put, for $n = e + 4$,

$$w = 2^{-n}, \quad h = 2^{-4n}, \quad \delta_e = 2^{-8n}, \quad r_e = 2^{-9n}, \quad K_e = 1/r_e.$$

Place $K_e$ addresses on a $2^{3n}$-by-$2^{3n}$-by-$2^{3n}$ grid of spacing $h$ in the cube beginning at $(4w, 0, 0)$. Its side is less than $w$. The proof of Lemma 12.1 applies verbatim: squared address separation is at least $\max(\delta_i, \delta_j)$, the coordinate bounds are unchanged, and their stereographic unit vectors obey

$$1 - (u_i^t u_j)^2 \ge \delta_i/2 \quad (i \ne j), \qquad r_e/\delta_e = w. \tag{17}$$

For rational $t \in [0,1]$ define

$$a(t) = \frac{2t}{1+t^2}, \quad b(t) = \frac{1-t^2}{1+t^2}, \quad d(t) = a(t)^2.$$

Then $a^2 + b^2 = 1$, $d$ increases strictly from zero to one, and

$$a(t) - a(s) = \frac{2(t-s)(1-st)}{(1+s^2)(1+t^2)}, \qquad 0 \le d(t) - d(s) \le 4(t-s) \quad (s \le t).$$

For $i = (e,k)$ put $t_i = (k+1)r_e$, $d_i = d(t_i) > 0$, and

$$p_i = J\big(\max(1/2, \min(1, \alpha_e/d_i))\big).$$

These numbers are uniformly left-c.e. by the rational formula and monotonicity of $J$. One fixed ordinary program emits the rational unit rays $(au_i, b)$ for all nonnegative rational circle points with $a^2 < p_i$ whenever that comparison is witnessed, and also the default ray $(0,0,0,0,1)$. Rational circle density gives own-address supremum $p_i$, including zero via the default. Encode each into five dyadic fields as in Theorem 12.4, and use the seven-coordinate verifier with $D = 8$. False histories have overlap at most $5/69$; any rival address has overlap at most $1 - \delta_i/2$ against $u_i$. The default is a nonzero ray, so Lemma 15.1 applies. The graph, initial logical direction, enumerator, read and all gates are fixed independently of the eventual preparation.

Given an ordinary index for target $\alpha$, use $0'$ to decide positivity. If it is zero return the reject-on-launch basis start described below. Otherwise find rational $0 < l < \alpha$ and choose a padded index $e$ for this same target so large that $2^{-n} < l/32$. With $0'$ choose the largest $k \in \{0, \dots, K_e - 1\}$ satisfying $d(kr_e) < \alpha$. Finite binary search works, since $d(0) < \alpha$ and $d(1) = 1 \not< \alpha$. At an internal equality this picks the preceding bin; at $\alpha = 1$ it picks the last bin, without testing equality to one. Thus

$$d(kr_e) < \alpha \le d_i, \qquad 0 \le d_i - \alpha < 4r_e.$$

For $\beta = \alpha/d_i$, (17) implies

$$1 - \beta < 4r_e/l < \delta_e/8, \qquad p_i = J(\beta) > 1 - (31/120)\delta_e > 1 - \delta_e/2.$$

In particular $\beta > 1/2$ and $p_i > 5/69$. Own rays therefore dominate every rival and false history. The exact projective supremum is $p_i$, so the work value is $G(p_i) = \beta$.

Add a one-time initialization with seven reference coordinates and one reject-on-launch live coordinate. Its mandatory launch sends a reference vector isometrically to reference tensored with reset reservoir, and rejects the extra coordinate; the start itself lies in the live space, not in a halting outcome. Other letters reject this initial phase and work never returns to it. Prepare

$$\psi_i = a(t_i)u_i \ \oplus \ b(t_i),$$

padding $u_i$ with zeros. This is a nonnegative rational unit vector on four reference coordinates and the reject-on-launch coordinate. The launch weight is $d_i$, giving exact value $d_i\beta = \alpha$. At target one $t_i = 1$ and the reject-on-launch amplitude is zero. Rational projection completion and the binary selector preserve these five starting coordinates inside copy zero and give the asserted continuing forms. Every rational start has a left-c.e. value, proving the exact spectrum.

All selector steps use $0'$ and finite ordinary arithmetic; the fixed enumerator itself has no oracle. Conversely any exact selector, even one allowed to return arbitrary rational starts of this apparatus, decides ordinary halting: use the name zero until halting and $1/2$ thereafter, and apply the reachable-span zero test in Theorem 11.5. Thus its least degree is $0'$.

For value-one hardness take an arbitrary ordinary index $j$, use a fixed tag and the last bin $k = K_e - 1$. Its $d_i = 1$, so prepare reference $u_i$ with no reject-on-launch amplitude, computably from $j$. If $\alpha_j = 1$, own overlaps approach one and the value is one. If $\alpha_j < 1$, own overlap supremum is below one, every rival is bounded by $1 - \delta_e/2$, and false histories by $5/69$. The full supremum is strictly below one, so $G$ gives value below one. The predicate $\alpha_j = 1$ is $\Pi^0_2$-complete by the progress-counter names of Theorem 13.2, and rational finite-word evaluation supplies membership. This proves completeness with fixed gates. □

On these hard starts value one has no perfectly accepting finite-word witness. Every own emitted circle point uses the strict bound $a^2 < p_i \leq 1$, and each rival and false history also has overlap below one. Each completed round therefore has ratio strictly below one by $G(p) < 1$; a finite word has finitely many rounds, and incomplete phases add no acceptance. The hard starts use the same four positive reference coordinates. Thus their endpoint predicate is not determined by coordinate support, which is the step that supplied the universal fixed-PFA $0''$ lower bound in Theorem 14.2. No public priority claim is made for this construction.

## How this work was done

The historical account in the following paragraphs is retained from the released version 1.1.0 and describes that version's development. The paragraph headed Version 2.0.0 describes the present additions.

**AI note.** The authors chose the question, directed the work and judged every result. AI models from two developers assisted. The question, whether a single rational probabilistic finite automaton can have a noncomputable value and whether the values are exactly the left-c.e. reals, was posed to OpenAI's GPT-6.1 Sol, with the theorem of Hochman and Meyerovitch as the model to aim at. In a long working session, several agents of that model family proposed routes, attacked them and audited one another. GPT-6.1 Sol derived the construction and the proofs of Theorems A, B and C, Corollary D, Proposition 8.1(i),(ii) and Propositions 10.1 and 10.4, posed the question of three states, and recorded the routes that failed, among them that uniform undecidability does not by itself produce one noncomputable value, and that summing rewards over witnessed computations cannot prevent a profitable certificate from being repeated. Before a result was kept, an agent that had not written it re-derived it refutation-first; the range theorem was re-derived independently by three agents, and the bound on the number of states by two. The literature lookups of the session were read-only.

Anthropic's Claude Opus 5.5 then checked the result refutation-first, as a review from the other developer. It re-derived each step from the notes of the session, before reading the session's own literature notes, and found no gap. It identified the compilation of an output machine into counter programs, and the universal program used for the bound on the number of states, as the steps written at the level of a sketch, and it noted that the rule for accepting and rejecting appeared in slightly different, equivalent forms across the notes. It judged the bookkeeping of the split test the most delicate step specific to the construction, and implemented the checking automaton and the payout for one small counter program as an explicit finite stochastic transition function in exact rational arithmetic; this test is described

under Verification status. A separate run of Claude Opus 5.5 searched the literature for prior results, and another run wrote this manuscript. In writing, Section 3 and the index check of Section 7 were written out in full and one form of the payout was fixed. The following were added in writing: Remarks 6.4 and 6.5; the reading of Theorem C as undecidability of strict emptiness for $A_*$; the bound on the number of states in Proposition 8.1(iii) and the remark after Proposition 8.1; Proposition 10.5; and, in Section 10, the questions on fixed matrices and probabilistic Büchi automata, a question on channels that the later review answered (below), and the remark on quantum automata (replaced in version 1.1.0 by Section 9). A separate agent of Claude Opus 5.5 that had not written the text read it as a hostile referee, checking every statement against its proof and every attribution against the literature notes. It found one error in a stated lemma: the parser of Lemma 4.4, as first written, accepted certificates with repeated final rows. The error did not affect any bound, since such rows only lower the probability that the joint test survives, and it is corrected in this version, together with the attributions and wordings that the reading flagged.

A fresh session of GPT-6.1 Sol then reviewed the written manuscript refutation-first, without the notes of the original session. It re-derived Theorems A–D from the text and found no error in them. It observed that the question on channels in the first draft is answered by Theorem C together with [11, Theorem 1]; that observation is now Corollary 10.7, which Claude Opus 5.5 checked against the primary source. It also flagged the attribution to Boche, Schaefer and Poor, the scope of the remark on quantum automata, the abstract's description of the initial distribution and several wordings, all corrected here.

Models are not listed as authors; the authors take responsibility for every claim. The proofs are given in full, and none of them uses a computation.

The quantum section of version 1.1.0 was derived by GPT-6.1 Sol in a separate research run. Claude Opus 5.5 checked its claims refutation-first, including an independently written exact rational toy two-way automaton. GPT-6.1 Sol then wrote Section 9, expanding the proofs of the included one-way subclasses and of the rational control-language realization, and wrote a new exact checker for the latter; Claude Opus 5.5 read the written section back refutation-first, re-deriving each statement and rerunning the checker. The two-way compiler was specified in prose in the research run; the toy engine did not instantiate that compiler. Those two-way statements were consequently deferred in version 1.1.0. The present version proves a one-way result by a different construction.

**Version 2.0.0.** The assistance named on the first page was used in developing and checking the mathematical arguments of this version. Proposition 9.8 and the results of Sections 11 to 15 were checked refutation-first by models of a second developer, which re-derived the arguments and ran independent exact finite checks. A separate blind attempt on the questions of Sections 11 to 14 proved the upper bounds but did not rediscover the constructions; after reading them it re-derived Theorems 11.5, 12.2, 12.4, 13.2 and 14.2. The closed-interval selector of Section 15 was also given to a separate review, which first attempted it blind without rediscovering the construction and then, after reading it, proved all three of its parts. The written text of this version, including the steps added in writing, such as the local successor rule of Section 11.2 and Lemma 11.3, was read back refutation-first by a model of the second developer. The programs described under Verification status check finite instances, constants and gadget identities in exact arithmetic; the infinite constructions rest on the proofs in the text. The authors take responsibility for the proofs, their scope and the use of the cited results.

## Verification status

Theorems A, B and C, Corollary D and Proposition 8.1(i),(ii) were checked analytically, at the level of the notes of the working session, by the re-derivations within the developing model family and by the refutation-first review from the other developer described above; Propositions 10.1 and 10.4 were checked by the same review in passing. The written proofs rest on Sections 3 and 7, whose written form, like the statements added in writing listed above, was checked by the hostile reading of the text, by a model of the family that wrote it, and then by the review of the written manuscript from the developer of the model that found the construction, described above. That review also tested the certificate format as written here, without control-state letters and with the corrected parser, in exact rational arithmetic: on one two-counter program and five choices of parameters it checked every reachable transition row, the masses of both tests, the gap on dishonest certificates, the payout and restart, the per-round decomposition on adversarial words and the repetition formula, with a planted dishonest certificate as a positive control, and in two exhaustive extensions it compared all 915,543 well-formed certificates in a fixed finite box (7 honest, 915,536 dishonest) against the full stochastic model, with no failures. Like the earlier test, it does not test the compilation of Section 3 or the universal program.

The finite-state implementation was tested in exact rational arithmetic during the review, before this text was written. The tested automaton has the payout (P3) and the index check (I1)–(I4) in the form written here; its certificates also list the control state of each row, which the format of Definition 4.2 omits. One test program had two counters, and its halting runs output exactly the binary words with an even number of ones; with the index check of Section 7 at $e = 1$, those with an odd number. The checking automaton, both tests with their slots, the register, the payout and the restart were built as an explicit transition function for five choices of $(B, a, b)$ and of the index check, namely $(4, 1/8, 1/2)$ and $(16, 1/8, 1/2)$ without it, $(4, 1/8, 1/2)$ with $e = 0$ and with $e = 1$, and $(3, 1/5, 3/5)$ with $e = 2$; these parameters need not satisfy (1), which the test does not use. Every transition row that was used was nonnegative with sum exactly 1. For each choice, on all valid certificates with at most four output bits (15 or 16 per choice), the surviving probabilities of the two tests were equal and agreed with the products of Lemma 4.5, and the probabilities entering yes and no agreed with Lemma 5.1. On 83 to 95 certificates per choice that were well formed but not valid, including certificates with a wrong initial index, $\Psi \geq \kappa_B \Phi$ held. On 150 further words per choice, mixing up to four certificates with truncations and inserted or appended letters, the acceptance probability equalled the sum in the proof of Lemma 5.2 and was at most the largest ratio $\mathrm{acc}(t)/(\mathrm{acc}(t) + \mathrm{rej}(t))$ of its certificates, and the formula of Lemma 5.3 held for $m = 1, 2, 5$. There were no failures. About 1,400 states outside $\{\mathsf{yes}, \mathsf{no}\}$ were reachable for each choice. The test checks the finite-state bookkeeping on one program; it does not test the compilation of Section 3 or the universal program, and no proof uses it. Reports of the checks are available from the authors.

For Section 9, the cross-family review used *PROVED* for claims it re-derived analytically and *COMPUTED* for its independent exact finite checks. Its priority assessment for the control-language range statement was *PARTLY KNOWN*: exact PFA simulation is folklore, and the value statement follows from Theorem A; the rational-unitary refinement was not found in print. The history-checking technique behind the two-way extension proposed in version 1.1.0, and the undecidability of isolated-cutpoint emptiness for 1.5-way automata, are known, due to Amano and Iwama; see Blondel, Jeandel, Koiran and Portier [4, p. 1464]. That extension, deferred there, is superseded by Theorem 11.5, whose one-way machines never reverse the

head. Propositions 9.1 and 9.2 were added in writing at the review's recommendation and are immediate, respectively, from [9] and from the folklore simulation [1, Theorem 4.2] with Theorem A; we claim no priority for them. The review re-derived Propositions 9.5 and 9.7 and Lemma 9.6 (*PROVED*); no priority search was made for them. These labels describe that review's findings, not an exhaustive priority claim.

The accompanying `quantum_checks.py`, added in version 1.1.0, checks the rational control-language construction directly from its unitary matrices, using exact fractions and an outcome/controller dynamic program independent of the expected PFA formulas, and the classical channel embedding of Proposition 9.2 on 63 words. It verified 4,812 column inner products and 1,929 acceptance comparisons with no failures, and detected a planted nonunitary coin and a planted incorrect sampler. Its detailed counts and positive controls are in `quantum_checks.txt`. It checks finite instances of the included constructions; it does not instantiate or certify a two-way counter-machine compiler, and none of the proofs depends on it. Its test automata already have deterministic starts, so the opening-letter step is not exercised. For the new one-way construction, one of the refutation-first checks instantiated the history verifier end to end for the emitting machine of Section 11.2, including its register graph, phase-local rational KW letters and certified run-length descriptions of actual finite words. The universal interpreter, the binary selector at that scale and the full dense orthogonal matrices were not instantiated. The checks also verified projection and rational-completion gadgets in exact arithmetic. Those are corroborating finite computations, not substitutes for the proofs in Section 11. The accompanying `kw_checks.py` checks the local successor rule of Section 11.2 against an independently written global successor, the actual linear register updates on honest and false histories of its emitting machine, and the inverse-payoff and preparation-weight identities of Section 15, with a planted wrong length guess as a positive control. Its results and limits are recorded in `kw_checks.txt`. The retained `quantum_checks.py` is unchanged from version 1.1.0; rerun for this version, it reproduces its recorded output `quantum_checks.txt`. The final PDF is built with three pdfLaTeX passes, without BibTeX or shell escape, and its final log is checked for warnings and box defects. None of these computations is used as a premise of a mathematical proof.

## Credit

Probabilistic finite automata are due to Rabin [26]. The verification of counter-machine runs by unary coin-flipping tests is due to Condon and Lipton [8], building on an equality test of Freivalds [14]; we follow the accounts of Madani, Hanks and Condon [19, 20] and Rote [27]. The undecidability of emptiness is due to Nasu and Honda [24]; it is often credited to Paz's book [25], and [27] gives its history. The simulation of Turing machines by counter machines is Minsky's [22, 23]. Reals defined from a computably enumerable undecidable set, as $\alpha_*$ is, go back to Specker [30]. The model of a range theorem is that of Hochman and Meyerovitch [16]. Quantum automata with control language are due to Bertoni, Mereghetti and Palano [3, 21], and measure-many automata to Kondacs and Watrous [17]; the compact-closure method for measure-once automata is that of Derksen, Jeandel and Koiran [9]; exact simulation of probabilistic by quantum automata is folklore [1, Theorem 4.2]. For the additions of version 2.0.0: undecidability of cutpoint isolation for probabilistic automata is due to Bertoni [33] and Bertoni, Mauri and Torelli [34], its $\Sigma_2^0$-completeness to Chadha, Sistla and Viswanathan [36], and undecidability of value one to Gimbert and Oualhadj [15]; fixed matrices with the initial distribution as input were studied by Rote [28]; the symmetric read

uses the swap identity of Buhrman, Cleve, Watrous and de Wolf [35]; and the Latvian remark uses Kribs [39], Gao and Rouzé [38], Mereghetti, Palano and Raucci [40] and Ambainis et al. [32].

## References


[1] A. Ambainis and A. Yakaryılmaz, *Automata and quantum computing*, arXiv:1507.01988 (2015; version 2, 2018), https://arxiv.org/abs/1507.01988.

[2] C. Baier, N. Bertrand and M. Größer, *On decision problems for probabilistic Büchi automata*, in *Foundations of Software Science and Computational Structures (FoSSaCS 2008)*, Lecture Notes in Computer Science **4962**, Springer, 2008, 287–301.

[3] A. Bertoni, C. Mereghetti and B. Palano, *Quantum computing: 1-way quantum automata*, in *Developments in Language Theory (DLT 2003)*, Lecture Notes in Computer Science **2710**, Springer, 2003, 1–20.

[4] V. D. Blondel, E. Jeandel, P. Koiran and N. Portier, *Decidable and undecidable problems about quantum automata*, SIAM Journal on Computing **34** (2005), 1464–1473.

[5] H. Boche, R. F. Schaefer and H. V. Poor, *Shannon meets Turing: non-computability and non-approximability of the finite state channel capacity*, Communications in Information and Systems **20** (2020), 81–116. arXiv:2008.13270.

[6] C. S. Calude, P. H. Hertling, B. Khoussainov and Y. Wang, *Recursively enumerable reals and Chaitin Ω numbers*, Theoretical Computer Science **255** (2001), 125–149.

[7] G. J. Chaitin, *A theory of program size formally identical to information theory*, Journal of the ACM **22** (1975), 329–340.

[8] A. Condon and R. J. Lipton, *On the complexity of space bounded interactive proofs (extended abstract)*, in *30th Annual Symposium on Foundations of Computer Science*, IEEE, 1989, 462–467.

[9] H. Derksen, E. Jeandel and P. Koiran, *Quantum automata and algebraic groups*, Journal of Symbolic Computation **39** (2005), 357–371, doi:10.1016/j.jsc.2004.11.008.

[10] R. G. Downey and D. R. Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer, New York, 2010.

[11] D. Elkouss and D. Pérez-García, *Memory effects can make the transmission capability of a communication channel uncomputable*, Nature Communications **9** (2018), 1149.

[12] N. Fijalkow, *Undecidability results for probabilistic automata*, ACM SIGLOG News **4**(4) (2017), 10–17.

[13] N. Fijalkow, H. Gimbert, E. Kelmendi and Y. Oualhadj, *Deciding the value 1 problem for probabilistic leaktight automata*, Logical Methods in Computer Science **11**(2) (2015), paper 12. arXiv:1504.04136.

[14] R. Freivalds, *Probabilistic two-way machines*, in *Mathematical Foundations of Computer Science 1981*, Lecture Notes in Computer Science **118**, Springer, 1981, 33–45.

[15] H. Gimbert and Y. Oualhadj, *Probabilistic automata on finite words: decidable and undecidable problems*, in *Automata, Languages and Programming (ICALP 2010)*, Lecture Notes in Computer Science **6199**, Springer, 2010, 527–538.

[16] M. Hochman and T. Meyerovitch, *A characterization of the entropies of multidimensional shifts of finite type*, Annals of Mathematics **171** (2010), 2011–2038.

[17] A. Kondacs and J. Watrous, *On the power of quantum finite state automata*, in *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS 1997)*, IEEE, 1997, 66–75.

[18] A. Kučera and T. A. Slaman, *Randomness and recursive enumerability*, SIAM Journal on Computing **31** (2001), 199–211.

[19] O. Madani, S. Hanks and A. Condon, *On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems*, in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, AAAI Press, 1999, 541–548.

[20] O. Madani, S. Hanks and A. Condon, *On the undecidability of probabilistic planning and related stochastic optimization problems*, Artificial Intelligence **147** (2003), 5–34.

[21] C. Mereghetti and B. Palano, *Quantum finite automata with control language*, RAIRO–Theoretical Informatics and Applications **40** (2006), 315–332, doi:10.1051/ita:2006007.

[22] M. L. Minsky, *Recursive unsolvability of Post's problem of "Tag" and other topics in theory of Turing machines*, Annals of Mathematics **74** (1961), 437–455.

[23] M. L. Minsky, *Computation: Finite and Infinite Machines*, Prentice-Hall, Englewood Cliffs, 1967.

[24] M. Nasu and N. Honda, *Mappings induced by PGSM-mappings and some recursively unsolvable problems of finite probabilistic automata*, Information and Control **15** (1969), 250–273.

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

[26] M. O. Rabin, *Probabilistic automata*, Information and Control **6** (1963), 230–245.

[27] G. Rote, *Probabilistic finite automaton emptiness is undecidable*, 2024. arXiv:2405.03035.

[28] G. Rote, *Probabilistic finite automaton emptiness is undecidable for a fixed automaton*, in *50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, Leibniz International Proceedings in Informatics **345**, 2025, paper 86. arXiv:2412.05198.

[29] R. I. Soare, *Turing Computability: Theory and Applications*, Springer, Berlin, 2016.

[30] E. Specker, *Nicht konstruktiv beweisbare Sätze der Analysis*, Journal of Symbolic Logic **14** (1949), 145–158.

[31] A. M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society (2) **42** (1936–37), 230–265.

[32] A. Ambainis, M. Beaudry, M. Golovkins, A. Kikusts, M. Mercer and D. Thérien, *Algebraic results on quantum automata*, Theory of Computing Systems **39** (2006), 165–188, doi:10.1007/s00224-005-1263-x.

[33] A. Bertoni, *The solution of problems relative to probabilistic automata in the frame of the formal languages theory*, in *GI – 4. Jahrestagung*, Lecture Notes in Computer Science **26**, Springer, 1975, 107–112.

[34] A. Bertoni, G. Mauri and M. Torelli, *Some recursively unsolvable problems relating to isolated cutpoints in probabilistic automata*, in *Automata, Languages and Programming (ICALP 1977)*, Lecture Notes in Computer Science **52**, Springer, 1977, 87–94, doi:10.1007/3-540-08342-1_7.

[35] H. Buhrman, R. Cleve, J. Watrous and R. de Wolf, *Quantum fingerprinting*, Physical Review Letters **87** (2001), 167902, https://arxiv.org/abs/quant-ph/0102001.

[36] R. Chadha, A. P. Sistla and M. Viswanathan, *Probabilistic automata with isolated cut-points*, in *MFCS 2013*, Lecture Notes in Computer Science **8087**, Springer, 2013, 254–265, doi:10.1007/978-3-642-40313-2_24.

[37] N. Fijalkow, *What is known about the Value 1 Problem for Probabilistic Automata?*, arXiv:1410.3770 (2014), https://arxiv.org/abs/1410.3770.

[38] L. Gao and C. Rouzé, *Complete entropic inequalities for quantum Markov chains*, Archive for Rational Mechanics and Analysis **245** (2022), 183–238, doi:10.1007/s00205-022-01785-1; https://arxiv.org/abs/2102.04146.

[39] D. W. Kribs, *Quantum channels, wavelets, dilations and representations of $\mathcal{O}_n$*, Proceedings of the Edinburgh Mathematical Society **46** (2003), 421–433, doi:10.1017/S0013091501000980.

[40] C. Mereghetti, B. Palano and P. Raucci, *Latvian quantum finite state automata for unary languages*, Electronic Proceedings in Theoretical Computer Science **388** (2023), 63–78, doi:10.4204/EPTCS.388.8; https://arxiv.org/abs/2309.08720.

[41] H. Rogers, Jr., *Theory of Recursive Functions and Effective Computability*, McGraw–Hill, New York, 1967.

[42] R. I. Soare, *Recursively Enumerable Sets and Degrees*, Springer, Berlin, 1987.

[43] K. Weihrauch, *Computable Analysis: An Introduction*, Springer, Berlin, 2000.

[44] A. Yakaryılmaz and A. C. C. Say, *Languages recognized by nondeterministic quantum finite automata*, Quantum Information and Computation **10** (2010), 747–770, https://arxiv.org/abs/0902.2081.