On the Compressed-Oracle Technique, and Post-Quantum Security of Proofs of Sequential Work [⋆]

Kai-Min Chung[1], Serge Fehr[2,3], Yu-Hsuan Huang[4], and Tai-Ning Liao[5]

[1] Academia Sinica, Taiwan (kmchung@iis.sinica.edu.tw)
[2] CWI, Amsterdam, Netherlands (serge.fehr@cwi.nl)
[3] Mathematical Institute, Leiden University, Netherlands
[4] National Chiao-Tung University, Taiwan (asd00012334.cs04@nctu.edu.tw)
[5] National Taiwan University, Taiwan (tonyliao8631@gmail.com)

Abstract. We revisit the so-called compressed oracle technique, introduced by Zhandry for analyzing quantum algorithms in the quantum random oracle model (QROM). To start off with, we offer a concise exposition of the technique, which easily extends to the parallel-query QROM, where in each query-round the considered algorithm may make several queries to the QROM in parallel. This variant of the QROM allows for a more fine-grained query-complexity analysis.

Our main technical contribution is a framework that simplifies the use of (the parallel-query generalization of) the compressed oracle technique for proving query complexity results. With our framework in place, whenever applicable, it is possible to prove quantum query complexity lower bounds by means of purely classical reasoning. More than that, for typical examples the crucial classical observations that give rise to the classical bounds are sufficient to conclude the corresponding quantum bounds.

We demonstrate this on a few examples, recovering known results but also obtaining new results. Our main target is the hardness of finding a q-chain with fewer than q parallel queries, i.e., a sequence $x_0, x_1, \ldots, x_q$ with $x_i = H(x_{i-1})$ for all $1 \leq i \leq q$.

The above problem of finding a hash chain is of fundamental importance in the context of proofs of sequential work. Indeed, as a concrete cryptographic application of our techniques, we prove quantum security of the Simple Proofs of Sequential Work by Cohen and Pietrzak.

1 Introduction

Background. The random oracle (RO) methodology [2], which treats a cryptographic hash function $H : \{0,1\}^n \to \{0,1\}^m$ as an external *oracle*, has proven

[⋆] This research is partially supported by Ministry of Science and Technology, Taiwan, under Grant no. MOST 109-2223-E-001 -001 -MY3, MOST QC project, under Grant no. MOST 109-2627-M-002-003 -, and Executive Yuan Data Safety and Talent Cultivation Project (AS-KPQ-110-DSTCP).

to be a successful way to design very efficient cryptographic protocols and arguing them secure in a rigorous yet idealized manner. Even though it is known that in principle the methodology can break down [7] and a “proven secure” protocol may become insecure in the actual (non-idealized) setting, experience has shown that for natural protocols this does not seem to happen.

In case of a *quantum* adversary that may locally run a quantum computer, the RO needs to be modeled as a quantum operation that is capable of answering queries *in superposition*, in order to reasonably reflect the capabilities of an attacker in the non-idealized setting [5]. This is then referred to as the *quantum random oracle model* (QROM). Unfortunately, this change in the model renders typical RO-security proofs invalid. One reason is that in the ordinary RO model the security reduction can inspect the queries that the adversary makes to the RO, while this is not possible anymore in the quantum setting when the queries are quantum states in superposition — at least not without disturbing the query state significantly and, typically, uncontrollably.

The Compressed Oracle. A very powerful tool to deal with the QROM is the so-called *compressed oracle* technique, introduced by Zhandry [20]. On a conceptual level, the technique very much resembles the classical "lazy sampling" technique; on a technical level, the idea is to consider a *quantum purification* of the random choice of the function H, and to analyze the internal state of the RO then in the Fourier domain.

This idea has proven to be very powerful. On the one hand, it gave rise to new and shorter proofs for known lower bound results on the query complexity of quantum algorithms (like Grover [13,3]); on the other hand, it enabled to prove new cryptographic security results, like in the context of *indifferentiability* [20,11], or, more recently, the *Fiat-Shamir transformation* [17], when considering a quantum adversary. However, it still is quite cumbersome to actually employ the compressed oracle technique; proofs tend to be hard to read, and they require a good understanding of quantum information science.

Our Results. We first present a *concise* yet *mathematically rigorous* exposition of the compressed oracle technique. Our exposition differs from other descriptions (e.g. [20,15,11,8,14]) in that we adopt a more abstract view.

We also consider a generalization of the compressed-oracle technique to the *parallel-query* QROM. In this variation, the considered quantum oracle algorithm may make *several* queries to the QROM *in parallel* in each query-round. The main difference between parallel and sequential queries is of course that sequential queries may be *adaptive*, i.e., the queried value x may depend on the hash learned in a previous query, while parallel queries are limited to be *non-adaptive*. This variation of the QROM allows for a more fine-grained query-complexity analysis that distinguishes between the number q of query rounds, and the number k of queries made *per round*; the *total* number of queries made is then obviously given by $Q = kq$. This way of studying the query complexity of quantum oracle algorithms is in particular suited for analyzing how well a computational task can or cannot be parallelized (some more on this below).

As our first main technical contribution, we propose an abstract framework that simplifies the use of (our generalized version of) the compressed oracle technique in certain cases. In particular, with our new framework in place and when applicable, it is possible to prove *quantum* query complexity lower bounds by means of purely *classical* reasoning: all the quantum aspects are abstracted away. This means that no knowledge about quantum information science is necessary in order to apply our framework. If applicable, the reasoning is purely by means of identifying some classical property of the problem at hand and applying our meta-theorems. More than that, the necessary classical property can typically be extracted from the — typically much simpler — proof for the classical bound.

We demonstrate the workings and the power of our framework on a few examples, recovering known and finding new bounds. For example, with q, k, m as above, we show that the success probability of finding a *preimage* is upper bounded by $O(kq^2/2^m)$, compared to the coarse-grained bound $O(Q^2/2^m)$ [3] that does not distinguish between sequential and parallel queries; this recovers the known fact that the naive way to parallelize a preimage search (by doing several executions of Grover [13] in parallel) is optimal [19]. We also show that the success probability of finding a *collision* is bounded by $O(k^2q^3/2^m)$, compared to the coarse-grained bound $O(Q^3/2^m)$ [1] that does not distinguish between sequential and parallel queries. Like for Grover, this shows optimality for the obvious parallelization of the BHT collision finding algorithm [6]. We are not aware of any prior optimality result on parallel collision search; [16] shows a corresponding bound for *element distinctness*, but that bound does not apply here when considering a hash function with many collisions. Finally, our main example application is to the problem of finding a *q-chain*, i.e., a sequence $x_0, x_1, \ldots, x_q$ with $x_i = H(x_{i-1})$ for all $1 \leq i \leq q$ (or, more generally, that $H(x_{i-1})$ satisfies some other relation with x_i). While classically it is well known and not too hard to show that q parallel queries are necessary to find a q-chain, there has been no proven bound in the quantum setting — at least not until very recently (see the recent-related-work paragraph below).[6] Here, we show that the success probability of finding a q-chain using *fewer* than q queries is upper bounded by $O(k^3q^3/2^m)$. The proof is by means of recycling an observation that is crucial to the classical proof and plugging it into the right theorem(s) of our framework.

The problem of producing a hash chain is of fundamental importance in the context of *proofs of sequential work* (PoSW); indeed, a crucial ingredient of a PoSW is a computational problem that is hard/impossible to parallelize. Following up on this, our second main technical contribution is to show that the "Simple Proofs of Sequential Work" proposed by Cohen and Pietrzak [10] remain secure against quantum attacks. One might hope that this is simply a matter of plugging in our bound on the chain problem; unfortunately, it is more complicated: the entire protocol needs to be analyzed in the light of a quantum attack, which requires substantial additional work. As a matter of fact, we enrich our framework with a "calculus" that facilitates the latter. In return, our proof

[6] The problem of finding a q-chain looks similar to the *iterated hashing* studied in [18]; however, a crucial difference is that the start of the chain, x_0, is freely chosen here.

of the quantum security of the PoSW scheme is purely classical, with no need to understand anything about quantum information science.

Related Work. Independently and concurrently to the preparation of our work, the hardness of finding a q-chain with fewer than q queries and the security of the Cohen and Pietrzak PoSW scheme [10] against quantum attacks have also been analyzed and tackled by Blocki, Lee and Zhou in [4]. Their bounds are comparable to ours, and both works are exploiting the compressed oracle idea; however, the actual derivations and the conceptual contributions are quite different. Indeed, Blocki *et al.*'s work is very specific to the q-chain problem and the PoSW scheme, while in our work we provide a *general* framework for proving *quantum* query complexity bounds by means of *classical* reasoning, opening the door to derive further quantum query complexity bounds.

In a similar spirit, Chiesa, Manohar and Spooner [8] also offer means to apply the compressed oracle technique using purely classical combinatorial reasoning. A major difference is that in our work we allow *parallel* queries (which is crucial for our PoSW application), which confronted us with the main technical challenges in our work. Our framework easily applies to the main application of the Chiesa *et al.* paper (post-quantum secure SNARGs), but not vice versa.

2 Warm-up: Proving Classical Query Complexity Bounds

In this section, we discuss lower bounds on the *classical* query complexity in the classical ROM for a few example problems. This serves as a warm-up and allows us to point that that, when it then comes to analyzing the *quantum* query complexity of these problems, it is simply a matter of recycling certain observations from the classical proofs and plugging them into our framework.

2.1 The Lazy-Sampling Technique

First, we briefly recall the *lazy sampling* technique for efficiently simulating the classical RO. Instead of choosing a uniformly random function $H : \mathcal{X} \to \mathcal{Y}$ and answering each query x to the RO as $y = H(x)$, one can build up the hash function H "on the fly". Introduce a special symbol $\perp$ and initiate D_0 to be the constant-$\perp$ function. Then, inductively for $i = 1, 2, \ldots$, on receiving the i-th query x_i, check if this query has been made before, i.e., if $\exists\, j < i : x_i = x_j$. If so then set $D_i := D_{i-1}$; else, choose a uniformly random $y_i \in \mathcal{Y}$ and set D_i to $D_i := D_{i-1}[x_i \mapsto y_i]$, where in general $D[x \mapsto y]$ is defined by $D[x \mapsto y](x) = y$ and $D[x \mapsto y](\bar{x}) = D(\bar{x})$ for $\bar{x} \neq x$.[7] In either case, answer the query then with $y_i = D_i(x_i)$. We refer to such a function $D_i : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ as a *database*.

As it is easy to see, the lazy-sampling only affects the "internal workings" of the RO; any algorithm making queries to the standard RO (which samples H

[7] We stress that we define $D[x \mapsto y]$ also for x with $D(x) \neq \perp$, which then means that D is *re*defined at point x; this will be useful later.

as a random function at the beginning of time), or to the lazy-sampled variant (which builds up $D_0, D_1, \ldots$ as explained above), cannot see any difference.

For below, it will be convenient to write D_i, the "update" of D_{i-1} in response to query x_i, as $D_i = D_{i-1}^{\circlearrowleft x_i}$. Note that since $D_i(x) = y_i$ is chosen in a randomized way, $D_{i-1}^{\circlearrowleft x_i}$ is a random variable, strictly speaking.

2.2 Proving Classical Lower Bounds

One important feature of the lazy-sampling technique is that it allows for an *efficient* simulation of the random oracle. In the work here, we are instead interested in the lazy sampling technique as a tool for proving query complexity lower bounds. Our goal here is to show that the well-understood classical reasoning is very close to the reasoning that our framework will admit for proving bounds in the quantum setting. In order to align the two, certain argumentation below may appear overkill given the simplicity of the classical case.

Finding a Preimage. We first consider the example of finding a preimage of H; say, without loss of generality, finding $x \in \mathcal{X}$ with $H(x) = 0$. Thus, let $\mathcal{A}$ be an algorithm making q queries to the random oracle and outputting some x at the end, with the goal of x being a zero-preimage. A first simple observation is the following: if in the lazy-sampling picture after q queries the built-up database $D_q : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ does not map $\mathcal{A}$'s output x to 0, then $H(x)$ is unlikely to vanish, where $H(x)$ is understood to be obtained by making one more query to the oracle, i.e., $H(x) = D_{q+1}(x)$. More formally, if p is the probability that $H(x) = 0$ when $\mathcal{A}$ is interacting with the standard oracle, and p' is the probability that $D_q(x) = 0$ when $\mathcal{A}$ is interacting with the lazy-sampled oracle, then $p \leq p' + 1/|\mathcal{Y}|$. Looking ahead, this trivial observation is the classical counterpart of Corollary 1 (originally by Zhandry) that we encounter later.

By the above observation, it is sufficient to bound $P[\exists\, x : D_q(x) = 0]$. Furthermore, setting $\mathsf{PRMG} := \{D : \mathcal{X} \to \mathcal{Y} \cup \{\perp\} \mid \exists\, x : D(x) = 0\}$, we can write

$$P[\exists\, x : D_q(x) = 0] = P[D_q \in \mathsf{PRMG}\,] \leq \sum_i P[D_i \in \mathsf{PRMG} \mid D_{i-1} \notin \mathsf{PRMG}\,]\,.$$

In order to align the reasoning here with our framework, we introduce the *classical transition capacity*

$$\big[\,\neg\mathsf{PRMG} \to \mathsf{PRMG}\,\big] := \max_{\substack{D \notin \mathsf{PRMG} \\ x \in \mathcal{X}}} P[D^{\circlearrowleft x} \in \mathsf{PRMG}\,]$$

as the maximal probability that a database $D : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ with *no* zero-preimage will be turned into a database *with* a zero-preimage as a result of a query. Combining the above observations, we obtain that

$$p \leq q \cdot \big[\,\neg\mathsf{PRMG} \to \mathsf{PRMG}\,\big] + \frac{1}{|\mathcal{Y}|}\,. \tag{1}$$

Looking ahead, this is the classical counterpart to Theorem 1 (with P_s set to PRMG), which is in terms of the (appropriately defined) *quantum* transition capacity $\llbracket \cdot \to \cdot \rrbracket$.

The reader probably already sees that $\big[\neg\mathsf{PRMG} \to \mathsf{PRMG}\big] = 1/|\mathcal{Y}|$, leading to the (well-known) bound $p \leq (q+1)/|\mathcal{Y}|$. However, in order to better understand the general reasoning, we take a more careful look at bounding this transition capacity. For every $D \notin \mathsf{PRMG}$ and $x \in \mathcal{X}$, we identify a "*local*" property $\mathsf{L}^{D,x} \subseteq \mathcal{Y}$ that satisfies

$$D[x \mapsto y] \in \mathsf{PRMG} \iff y \in \mathsf{L}^{D,x} \,;$$

therefore, $P[D^{\circlearrowleft x} \in \mathsf{PRMG}\,] \leq P\big[D[x \mapsto U] \in \mathsf{PRMG}\big] = P[U \in \mathsf{L}^{D,x}]$ where U is defined to be uniformly random in $\mathcal{Y}$. Here, we can simply set $\mathsf{L}^{D,x} := \{0\}$ and thus obtain $\big[\neg\mathsf{PRMG} \to \mathsf{PRMG}\big] = P[U = 0] = 1/|\mathcal{Y}|$ as claimed.

The point of explicitly introducing $\mathsf{L}^{D,x}$ is that our framework will offer similar connections between the *quantum* transition capacity $\llbracket \cdot \to \cdot \rrbracket$ and the purely classically defined probability $P[U \in \mathsf{L}^{D,x}]$. Indeed, by means of the very same choice of local property $\mathsf{L}^{D,x}$, but then applying Theorem 2, we obtain

$$\llbracket \neg\mathsf{PRMG} \to \mathsf{PRMG} \rrbracket \leq \max_{D,x} \sqrt{10 P\big[U \in \mathsf{L}^{D,x}\big]} \leq \sqrt{\tfrac{10}{|\mathcal{Y}|}} \,.$$

By Theorem 1, this implies that the success probability p of a *quantum* algorithm to find a preimage is bounded by

$$p \leq \Big(q \llbracket \neg\mathsf{PRMG} \to \mathsf{PRMG} \rrbracket + \tfrac{1}{\sqrt{|\mathcal{Y}|}}\Big)^2 \leq \Big(q\sqrt{\tfrac{10}{|\mathcal{Y}|}} + \tfrac{1}{\sqrt{|\mathcal{Y}|}}\Big)^2 = O\big(\tfrac{q^2}{|\mathcal{Y}|}\big) \,,$$

confirming the optimality of the quadratic speed-up of Grover.

Finding a Preimage with Parallel Queries. The above (classical and quantum) reasoning can be extended to the parallel query model, where with each interaction with the RO, a query algorithm can make k queries in one go. The lazy-sampling technique then works in the obvious way, with the function update $D_i := D_{i-1}^{\circlearrowleft \mathbf{x}_i}$ now involving a query *vector* $\mathbf{x}_i \in \mathcal{X}^k$. This then gives rise to $\big[\neg\mathsf{PRMG} \xrightarrow{k} \mathsf{PRMG}\big]$, and (1) generalizes accordingly. For $D \notin \mathsf{PRMG}$ and $\mathbf{x} \in \mathcal{X}^k$, we then identify a *family* of local properties $\mathsf{L}_1^{D,\mathbf{x}}, \ldots, \mathsf{L}_k^{D,\mathbf{x}} \subseteq \mathcal{Y}$ so that

$$D[\mathbf{x} \mapsto \mathbf{y}] \in \mathsf{PRMG} \iff \exists\, i : y_i \in \mathsf{L}_i^{D,\mathbf{x}} \,, \tag{2}$$

and therefore, by the union bound, $P[D^{\circlearrowleft \mathbf{x}} \in \mathsf{PRMG}\,] \leq \sum_i P[U \in \mathsf{L}_i^{D,\mathbf{x}}]$. Setting $\mathsf{L}_1^{D,\mathbf{x}} = \ldots = \mathsf{L}_k^{D,\mathbf{x}} := \{0\}$, we now get $\big[\neg\mathsf{PRMG} \xrightarrow{k} \mathsf{PRMG}\big] = kP[U = 0] = k/|\mathcal{Y}|$, showing a factor-$k$ increase in the bound as expected. More interesting is that Theorem 2 still applies, implying that for the quantum version we have

$$\llbracket \neg\mathsf{PRMG} \xrightarrow{k} \mathsf{PRMG} \rrbracket \leq \max_{D,\mathbf{x}} \sqrt{10 \sum_i P\big[U \in \mathsf{L}_i^{D,\mathbf{x}}\big]} \leq \sqrt{\tfrac{10k}{|\mathcal{Y}|}} \,.$$

Plugging this into Theorem 1, we then get the bound

$$p \leq \Big(q\sqrt{\tfrac{10k}{|\mathcal{Y}|}} + \tfrac{1}{\sqrt{|\mathcal{Y}|}}\Big)^2 = O\big(\tfrac{q^2k}{|\mathcal{Y}|}\big)\,,$$

showing optimality of running k parallel executions of Grover.

Finding a Chain (with Parallel Queries). Another example we want to discuss here, where we now stick to the parallel query model, is the problem of finding a $(q+1)$-chain, i.e., a sequence $x_0, x_1, \ldots, x_{q+1}$ with $H(x_{i-1}) \triangleleft x_i$, with no more than q (parallel) queries. Here, $\triangleleft$ refers to an arbitrary relation among the elements of $\mathcal{X}$ and $\mathcal{Y}$; typical examples are: $y \triangleleft x$ if $x = y$, or if y is a prefix of y, or if y is an arbitrary continuous substring of x. Below, we set $\mathcal{Y}^{\triangleleft x} := \{y \in \mathcal{Y} \,|\, y \triangleleft x\}$ and $T := \max_x |\mathcal{Y}^{\triangleleft x}|$.

Using the same kind of reasoning as above, we can argue that

$$p \leq \sum_{s=1}^{q} \big[\, \neg\mathsf{CHN}^s \xrightarrow{k} \mathsf{CHN}^{s+1} \,\big] + \frac{q+2}{|\mathcal{Y}|}\,,$$

where $\mathsf{CHN}^s = \{D \,|\, \exists\, x_0, x_1, \ldots, x_s \in \mathcal{X} : D(x_{i-1}) \triangleleft x_i \;\forall i\}$. Here, we will exploit that after s (parallel) queries, $D_s \in \mathsf{SZ}_{\leq ks} := \{D \,|\, |\{x | D(x) \neq \bot\}| \leq ks\}$. Thus, the above extends to

$$p \leq \sum_{s=1}^{q} \big[\, \mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{CHN}^s \xrightarrow{k} \mathsf{CHN}^{s+1} \,\big] + \frac{q+2}{|\mathcal{Y}|}\,, \tag{3}$$

with the (classical) transition capacity here given by $\max P[D^{\circlearrowleft \mathbf{x}} \in \mathsf{CHN}^{s+1}]$, maximized over all $D \in \mathsf{SZ}_{\leq k(s-1)} \setminus \mathsf{CHN}^s$ and $\mathbf{x} \in \mathcal{X}^k$. To control the considered transition capacity, for any D and any $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$, we introduce the following local properties $\mathsf{L}_i^{D,\mathbf{x}} \subseteq \mathcal{Y}$ with $i = 1, \ldots, k$:

$$\mathsf{L}_i^{D,\mathbf{x}} = \bigcup_{\substack{x \in \mathcal{X} \\ D(x) \neq \bot}} \mathcal{Y}^{\triangleleft x} \cup \bigcup_{j=1}^{k} \mathcal{Y}^{\triangleleft x_j}\,, \tag{4}$$

so that $y_i \in \mathsf{L}_i^{D,\mathbf{x}}$ if $y_i \triangleleft x$ for some $x \in \mathcal{X}$ with $D(x) \neq \bot$ or $x \in \{x_1, \ldots, x_k\}$. They satisfy the following condition, which is slightly weaker than (2) used above.

Lemma 1. $D[\mathbf{x} \mapsto \mathbf{r}] \notin \mathsf{CHN}^s \wedge D[\mathbf{x} \mapsto \mathbf{u}] \in \mathsf{CHN}^{s+1} \;\Rightarrow\; \exists\, i : r_i \neq u_i \wedge u_i \in \mathsf{L}_i^{D,\mathbf{x}}$.

Proof. Write $D_\circ$ for $D[\mathbf{x} \mapsto \mathbf{r}]$ and D' for $D[\mathbf{x} \mapsto \mathbf{u}]$. Assume that $D' \in \mathsf{CHN}^{s+1}$, and let $\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{s+1} \in \mathcal{X}$ be such a chain, i.e., so that $D'(\hat{x}_j) \triangleleft \hat{x}_{j+1}$ for $j = 0, \ldots, s$. Let $s_\circ$ be the smallest j so that $D_\circ(\hat{x}_j) \neq D'(\hat{x}_j)$; if $s_\circ \geq s$ (or no such j exists) then $D_\circ(\hat{x}_j) = D'(\hat{x}_j) \triangleleft \hat{x}_{j+1}$ for $j = 0, \ldots, s-1$, and thus $D_\circ \in \mathsf{CHN}^s$ and we are done. Therefore, we may assume $s_\circ < s$. Furthermore, since $D_\circ(\bar{x}) = D'(\bar{x})$ for $\bar{x} \notin \{x_1, \ldots, x_k\}$, we must have that $\hat{x}_{s_\circ} = x_i$ for some $i \in \{1, \ldots, k\}$, and therefore $r_i = D_\circ(x_i) = D_\circ(\hat{x}_{s_\circ}) \neq D'(\hat{x}_{s_\circ}) = D'(x_i) = u_i$. Also, we have that $u_i = D'(x_i) = D'(\hat{x}_{s_\circ}) \triangleleft \hat{x}_{s_\circ+1}$ where $\hat{x}_{s_\circ+1}$ is such that $D'(\hat{x}_{s_\circ+1}) \triangleleft \hat{x}_{s_\circ+2}$ and thus $\neq \bot$. The latter means that either $D(\hat{x}_{s_\circ+1}) \neq \bot$ or $\hat{x}_{s_\circ+1} \in \{x_1, \ldots, x_k\}$ (or both). In either case we have that $u_i \in \mathsf{L}_i^{D,\mathbf{x}}$.

Applied to $\mathbf{r} := D(\mathbf{x})$ so that $D[\mathbf{x} \mapsto \mathbf{r}] = D$, we obtain $P[D^{\circlearrowleft \mathbf{x}} \in \mathsf{CHN}^{s+1}] \leq \sum_i P[U \in \mathsf{L}_i^{D,\mathbf{x}}]$. Given that, for $D \in \mathsf{SZ}_{\leq k(s-1)}$, the set $\{x | D(x) \neq \bot\}$ is bounded in size by $k(s-1)$, and $|\mathcal{Y}^{\lhd x}|, |\mathcal{Y}^{\lhd x_j}| \leq T$, we can bound the relevant probability $P[U \in \mathsf{L}_i^{D,x}] \leq ksT/|\mathcal{Y}|$. Hence, the considered classical transition capacity is bounded by $k^2 sT/|\mathcal{Y}|$. By (3), we thus have $p = O(k^2 q^2 T/|\mathcal{Y}|)$, which is in line with the bound given by Cohen-Pietrzak [10].

Also here, our framework allows us to lift the above reasoning to the quantum setting by plugging the core elements of the above reasoning for the classical case into our framework. Concretely, choosing the local properties $\mathsf{L}_i^{D,\mathbf{x}}$ as above whenever $D \in \mathsf{SZ}_{\leq k(s-1)}$, and to be constant-false otherwise, Lemma 1 ensures that we can apply Theorem 3 to bound the *quantum* transition capacity as

$$[\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{CHN}^s \xrightarrow{k} \mathsf{CHN}^{s+1}]\!] \leq e \max_{\mathbf{x},D} \sum_i \sqrt{10 P\big[U \in \mathsf{L}_i^{D,\mathbf{x}}\big]} \leq ek\sqrt{\tfrac{10k(q+1)T}{|\mathcal{Y}|}},$$

where e is Euler's number. Plugging this into Theorem 1, we then get the bound

$$p \leq \Big(qek\sqrt{\tfrac{10k(q+1)T}{|\mathcal{Y}|}} + \tfrac{q+2}{|\mathcal{Y}|}\Big)^2 = O\big(\tfrac{q^3 k^3 T}{|\mathcal{Y}|}\big)$$

on the success probability of a quantum oracle algorithm in finding a $(q{+}1)$-chain with no more than q k-parallel queries. Recall, T depends on the considered relation $y \lhd x$; $T = 1$ if y is required to be equal to x, or a prefix of x, and $T = m - n$ if y and x are n- and m-bit strings, respectively, and y is required to be a continuous substring of x.

3 Notation

3.1 Operators and Their Norms

Let $\mathcal{H} = \mathbb{C}^d$ be a finite-dimensional complex Hilbert space. We use the standard bra-ket notation for covariant and contravariant vectors in $\mathcal{H}$, i.e., for column and row vectors $\mathbb{C}^d$. We write $\mathcal{L}(\mathcal{H}, \mathcal{H}')$ for the linear maps, i.e., operators (or matrices), $A : \mathcal{H} \to \mathcal{H}'$, and we use $\mathcal{L}(\mathcal{H})$ as a short hand for $\mathcal{L}(\mathcal{H}, \mathcal{H})$. We write I for the identity operator in $\mathcal{L}(\mathcal{H})$. It is understood that pure states are given by norm-1 ket vectors $|\psi\rangle \in \mathcal{H}$ and mixed states by density operators $\rho \in \mathcal{L}(\mathcal{H})$.

A (possibly) mixed state $\rho \in \mathcal{L}(\mathcal{H})$ is said to be *supported* by subspace $\mathcal{H}_\circ \subseteq \mathcal{H}$ if the support of the operator ρ lies in $\mathcal{H}_\circ$, or, equivalently, if any purification $|\Psi\rangle \in \mathcal{H} \otimes \mathcal{H}$ of ρ lies in $\mathcal{H}_\circ \otimes \mathcal{H}$. A state is said to be supported by a family of (orthonormal) vectors if it is supported by the span of these vectors.

We write $\|A\|$ for the *operator norm* of $A \in \mathcal{L}(\mathcal{H}, \mathcal{H}')$ and recall that it is upper bounded by the *Frobenius norm*. Special choices of operators in $\mathcal{L}(\mathcal{H})$ are *projections* and *unitaries*. We assume familiarity with these notions, as well as with the notion of an *isometry* in $\mathcal{L}(\mathcal{H}, \mathcal{H}')$.

If $\mathcal{H}_\circ$ is a subspace of $\mathcal{H}$ and $A \in \mathcal{L}(\mathcal{H}_\circ)$ then we can naturally understand A as a map $A \in \mathcal{L}(\mathcal{H})$ by letting A act as zero-map on any $|\psi\rangle \in \mathcal{H}$ that is orthogonal to $\mathcal{H}_\circ$. We point out that this does not cause any ambiguity in

$\|A\|$. Vice versa, for any $A \in \mathcal{L}(\mathcal{H})$ we can consider its restriction to $\mathcal{H}_\circ$. Here, we have the following. If $\mathcal{H} = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_m$ is a decomposition of $\mathcal{H}$ into orthogonal subspaces $\mathcal{H}_i \subseteq \mathcal{H}$, and $A \in \mathcal{L}(\mathcal{H})$ is such that its restriction to $\mathcal{H}_i$ is a map $\mathcal{H}_i \to \mathcal{H}_i$ and coincides with $B_i \in \mathcal{L}(\mathcal{H}_i)$ for any $i \in \{1, \ldots, m\}$, then $\|A\| = \max_i \|B_i\|$. This is a property we are exploiting multiple times, typically making a reference then to "basic properties" of the operator norm.

3.2 The Computational and the Fourier Basis

Let $\mathcal{Y}$ be a finite Abelian group of cardinality M, and let $\{|y\rangle\}_{y\in\mathcal{Y}}$ be an (orthonormal) basis of $\mathcal{H} = \mathbb{C}^M$, where the basis vectors are labeled by the elements of $\mathcal{Y}$. We refer to this basis as the *computational basis*, and we also write $\mathbb{C}[\mathcal{Y}]$ for $\mathcal{H} = \mathbb{C}^M$ to emphasize that the considered space is spanned by basis vectors that are labeled by the elements in $\mathcal{Y}$. Let $\hat{\mathcal{Y}}$ be the *dual group* of $\mathcal{Y}$, which is known to be isomorphic to $\mathcal{Y}$, and thus to have cardinality M as well. Up to some exceptions, we consider $\hat{\mathcal{Y}}$ to be an *additive* group; the neutral element is denoted $\hat{0}$. We stress that we treat $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ as disjoint sets, even though in certain (common) cases they are *naturally* isomorphic and thus considered to be equal. The *Fourier basis* $\{|\hat{y}\rangle\}_{\hat{y}\in\hat{\mathcal{Y}}}$ of $\mathcal{H}$ is defined by the basis transformations

$$|\hat{y}\rangle = \frac{1}{\sqrt{M}}\sum_{y} \hat{y}(y)^*|y\rangle \qquad \text{and} \qquad |y\rangle = \frac{1}{\sqrt{M}}\sum_{\hat{y}} \hat{y}(y)|\hat{y}\rangle\,, \tag{5}$$

where $(\cdot)^*$ denotes complex conjugation. With the above convention on the notation, we have $\mathbb{C}[\mathcal{Y}] = \mathbb{C}[\hat{\mathcal{Y}}] = \mathcal{H}$.An elementary property of the Fourier basis is that the operator in $\mathcal{L}(\mathbb{C}[\mathcal{Y}] \otimes \mathbb{C}[\mathcal{Y}])$ defined by $|y\rangle|y'\rangle \mapsto |y+y'\rangle|y'\rangle$ for $y, y' \in \mathcal{Y}$ acts as $|\hat{y}\rangle|\hat{y}'\rangle \mapsto |y\rangle|\hat{y}-\hat{y}'\rangle$ for $\hat{y}, \hat{y}' \in \hat{\mathcal{Y}}$.

We will also consider extensions $\mathcal{Y}\cup\{\perp\}$ and $\hat{\mathcal{Y}}\cup\{\perp\}$ of the sets $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ by including a special symbol $\perp$. We will then fix a norm-1 vector $|\perp\rangle \in \mathbb{C}^{M+1}$ that is orthogonal to $\mathbb{C}[\mathcal{Y}] = \mathbb{C}[\hat{\mathcal{Y}}]$, given a fixed embedding of $\mathbb{C}[\mathcal{Y}] = \mathbb{C}^M$ into $\mathbb{C}^{M+1}$. In line with our notation, $\mathbb{C}^{M+1}$ is then referred to as $\mathbb{C}[\mathcal{Y}\cup\{\perp\}] = \mathbb{C}[\hat{\mathcal{Y}}\cup\{\perp\}]$.

3.3 Functions and Their (Quantum) Representations

For an arbitrary but fixed non-empty finite set $\mathcal{X}$, we let $\mathfrak{H}$ be the set of functions $H : \mathcal{X} \to \mathcal{Y}$. Similarly, $\hat{\mathfrak{H}}$ denotes the set of functions $\hat{H} : \mathcal{X} \to \hat{\mathcal{Y}}$. Given that we can represent H by its function table $\{H(x)\}_{x\in\mathcal{X}}$, and $|y\rangle \in \mathbb{C}[\mathcal{Y}]$ is understood as a "quantum representation" of $y \in \mathcal{Y}$, we consider $|H\rangle = \bigotimes_x |H(x)\rangle$ to be the "quantum representation" of H, where in such a tensor product we implicitly consider the different registers to be *labeled* by $x \in \mathcal{X}$ in the obvious way. By our naming convention, the space $\bigotimes_x \mathbb{C}[\mathcal{Y}]$ spanned by all vectors $|H\rangle = \bigotimes_x |H(x)\rangle$ with $H \in \mathfrak{H}$ is denoted $\mathbb{C}[\mathfrak{H}]$. Similarly, for the "quantum representation" of $\hat{H} \in \hat{\mathfrak{H}}$ as$|\hat{H}\rangle = \bigotimes_x |\hat{H}(x)\rangle$. By applying (5) register-wise, any $|H\rangle$ is supported by vectors $|\hat{H}\rangle$ with $\hat{H} \in \hat{\mathfrak{H}}$, and vice versa. Thus, $\mathbb{C}[\mathfrak{H}] = \mathbb{C}[\hat{\mathfrak{H}}]$.

Extending $\mathcal{Y}$ to $\bar{\mathcal{Y}} := \mathcal{Y} \cup \{\perp\}$, we also consider the set $\mathfrak{D}$ of functions (referred to as *databases*) $D : \mathcal{X} \to \bar{\mathcal{Y}}$. In line with the above, we then obtain $|D\rangle = \bigotimes_x |D(x)\rangle \in \bigotimes_x \mathbb{C}[\bar{\mathcal{Y}}] = \mathbb{C}[\mathfrak{D}]$. We also consider the set $\hat{\mathfrak{D}}$ of functions $\hat{D} : \mathcal{X} \to \hat{\mathcal{Y}} \cup \{\perp\}$ and have $\mathbb{C}[\mathfrak{D}] = \mathbb{C}[\hat{\mathfrak{D}}]$.

For $D \in \mathfrak{D}$ and $\mathbf{x} = (x_1, \dots, x_k) \in \mathcal{X}^k$, we write $D(\mathbf{x})$ for $\big(D(x_1), \dots, D(x_k)\big)$ in $\bar{\mathcal{Y}}^k$; similarly for $H \in \mathfrak{H}$. Furthermore, for $\mathbf{x}$ with pairwise distinct entries and $\mathbf{r} = (r_1, \dots, r_k) \in \bar{\mathcal{Y}}^k$, we define $D[\mathbf{x} \mapsto \mathbf{r}] \in \mathfrak{D}$ to be the database with $D[\mathbf{x}\mapsto\mathbf{r}](x_i) = r_i$ and $D[\mathbf{x}\mapsto\mathbf{r}](\bar{x}) = D(\bar{x})\ \forall\, \bar{x} \notin \{x_1, \dots, x_k\}$.

4 Zhandry's Compressed Oracle - Refurbished

4.1 The Compressed Oracle

The core ideas of Zhandry's compressed oracle are, first, to consider a *superposition* $\sum_H |H\rangle$ of all possible functions $H \in \mathfrak{H}$, rather than a uniformly random choice; this *purified* oracle is indistinguishable from the original random oracle. Second, to then analyze the behavior of this purified oracle in the *Fourier* basis. Indeed, the initial state of the oracle is given by

$$|\Pi_0\rangle = \sum_H |H\rangle = \bigotimes_x \Big(\sum_y |y\rangle\Big) = \bigotimes_x |\hat{0}\rangle = |\hat{\mathbf{0}}\rangle \in \mathbb{C}[\mathfrak{H}]\,, \tag{6}$$

with $\hat{\mathbf{0}} \in \hat{\mathfrak{H}}$ the constant-$\hat{0}$ function. Furthermore, an oracle query invokes the unitary map O, given by

$$\mathsf{O} : |x\rangle|y\rangle \otimes |H\rangle \mapsto |x\rangle|y + H(x)\rangle \otimes |H\rangle$$

in the computational basis; in the Fourier basis, this becomes

$$\mathsf{O} : |x\rangle|\hat{y}\rangle \otimes |\hat{H}\rangle \mapsto |x\rangle|\hat{y}\rangle \otimes \mathsf{O}_{x\hat{y}}|\hat{H}\rangle = \ |x\rangle|\hat{y}\rangle \otimes |\hat{H} - \hat{y} \cdot \delta_x\rangle\,, \tag{7}$$

where the equality is the definition of $\mathsf{O}_{x\hat{y}}$, and $\delta_x : \mathcal{X} \to \{0,1\}$ satisfies $\delta_x(x) = 1$ and $\delta_x(x') = 0$ for all $x' \neq x$. Note that $\mathsf{O}_{x\hat{y}}$ acts on register x only, and $\mathsf{O}_{x\hat{y}}\mathsf{O}_{x\hat{y}'} = \mathsf{O}_{x,\hat{y}+\hat{y}'}$; thus, $\mathsf{O}_{x\hat{y}}$ and $\mathsf{O}_{x'\hat{y}'}$ all commute. As an immediate consequence of (6) and (7) above, the internal state of the oracle after q queries is supported by state vectors of the form $|\hat{H}\rangle = |\hat{y}_1\delta_{x_1} + \cdots + \hat{y}_q\delta_{x_q}\rangle$.

The actual *compressed* oracle (respectively some version of it) is now obtained by applying the isometry

$$\mathsf{Comp}_x = |\perp\rangle\langle\hat{0}| + \sum_{\hat{z}\neq\hat{0}} |\hat{z}\rangle\langle\hat{z}| : \mathbb{C}[\mathcal{Y}] \to \mathbb{C}[\bar{\mathcal{Y}}],\ |\hat{y}\rangle \mapsto \begin{cases} |\perp\rangle \text{ if } \hat{y} = \hat{0} \\ |\hat{y}\rangle \ \text{ if } \hat{y} \neq \hat{0} \end{cases}$$

to all registers $x \in \mathcal{X}$ (and then viewing the result in the computational basis). This "compression" operator $\mathsf{Comp} := \bigotimes_x \mathsf{Comp}_x : \mathbb{C}[\mathfrak{H}] \to \mathbb{C}[\mathfrak{D}]$ maps $|\Pi_0\rangle$ to

$$|\Delta_0\rangle := \mathsf{Comp}\,|\Pi_0\rangle = \Big(\bigotimes_x \mathsf{Comp}_x\Big)\Big(\bigotimes_x |\hat{0}\rangle\Big) = \bigotimes_x \mathsf{Comp}_x|\hat{0}\rangle = \bigotimes_x |\perp\rangle = |\perp\rangle,$$

which is the quantum representation of the trivial database $\perp$ that maps any $x \in \mathcal{X}$ to $\perp$. More generally, for any $\hat{H} \in \hat{\mathfrak{H}}$, $\mathsf{Comp}\,|\hat{H}\rangle = |\hat{D}\rangle$ where $\hat{D} \in \hat{\mathfrak{D}}$ is such that $\hat{D}(x) = \hat{H}(x)$ whenever $\hat{H}(x) \neq 0$, and $\hat{D}(x) = \perp$ whenever $\hat{H}(x) = 0$. Thus, the internal state of the compressed oracle after q queries is supported by vectors $|D\rangle$ in the computational basis (respectively $|\hat{D}\rangle$ in the Fourier basis) for which $D(x) = \perp$ (respectively $\hat{D}(x) = \perp$) for all but at most q choices of x.

This representation of the internal state of the purified random oracle is referred to as the *compressed* oracle because, for a bounded number of queries, these state vectors $|D\rangle$ can be efficiently represented and the evolution of the oracle then efficiently computed (see the full version [9]). In this work, we are not concerned with such a computational efficiency aspect.

4.2 Linking the Compressed and the Original Oracle

The following result (originally by Zhandry [20]) links the compressed oracle with the original standard oracle. Recall that $M = |\mathcal{Y}|$.

Lemma 2. *Consider an arbitrary normalized $|\Pi\rangle \in \mathbb{C}[\mathfrak{H}]$. Let $|\Delta\rangle = \mathsf{Comp}\,|\Pi\rangle$ in $\mathbb{C}[\mathfrak{D}]$ be the corresponding "compressed database". Let $\mathbf{x} = (x_1, \ldots, x_\ell)$ consist of pairwise distinct $x_i \in \mathcal{X}$, let $\mathbf{y} = (y_1, \ldots, y_\ell) \in \mathcal{Y}^\ell$, and set $P_\mathbf{x} := |y_1\rangle\langle y_1| \otimes \cdots \otimes |y_\ell\rangle\langle y_\ell|$ with the understanding that $|y_i\rangle\langle y_i|$ acts on register x_i. Then*

$$\|P_\mathbf{x}|\Pi\rangle\| \leq \|P_\mathbf{x}|\Delta\rangle\| + \sqrt{\frac{\ell}{M}}\,.$$

This translates to the following statement in terms of algorithmic language; rigorous proofs of both statements are given in the full version [9].

Corollary 1 (Zhandry). *Let $R \subseteq \mathcal{X}^\ell \times \mathcal{Y}^\ell$ be a relation. Let $\mathcal{A}$ be an oracle quantum algorithm that outputs $\mathbf{x} \in \mathcal{X}^\ell$ and $\mathbf{y} \in \mathcal{X}^\ell$. Let p be the probability that $\mathbf{y} = H(\mathbf{x})$ and $(\mathbf{x}, \mathbf{y}) \in R$ when $\mathcal{A}$ has interacted with the standard random oracle, initialized with a random function H. Similarly, let p' be the probability that $\mathbf{y} = D(\mathbf{x})$ and $(\mathbf{x}, \mathbf{y}) \in R$ when $\mathcal{A}$ has interacted with the compressed oracle instead and D is obtained by measuring its internal state. Then*

$$\sqrt{p} \leq \sqrt{p'} + \sqrt{\frac{\ell}{M}}\,.$$

4.3 Working Out the Transition Matrix

Here, we work out the matrix (in the computational basis) that describes the evolution that the compressed oracle undergoes as a result of an oracle query. For this, it is necessary to extend the domain $\mathbb{C}[\mathcal{Y}]$ of Comp_x to $\mathbb{C}[\bar{\mathcal{Y}}]$ by declaring that $\mathsf{Comp}_x|\perp\rangle = |\hat{0}\rangle$. This turns Comp_x into a *unitary* on $\mathbb{C}[\bar{\mathcal{Y}}]$, and correspondingly then for Comp. Formally, we are then interested in the unitary

$$\mathsf{cO} := \mathsf{Comp} \circ \mathsf{O} \circ \mathsf{Comp}^\dagger \in \mathcal{L}\big(\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{Y}] \otimes \mathbb{C}[\mathfrak{D}]\big)\,,$$

which maps $|x\rangle|\hat{y}\rangle \otimes |D\rangle$ to $|x\rangle|\hat{y}\rangle \otimes \mathsf{cO}_{x\hat{y}}|D\rangle$ for any $D \in \mathfrak{D}$, where the unitary $\mathsf{cO}_{x\hat{y}} := \mathsf{Comp}_x \circ \mathsf{O}_{x\hat{y}} \circ \mathsf{Comp}_x^\dagger \in \mathcal{L}(\mathbb{C}[\bar{\mathcal{Y}}])$ acts on the x-register only.

Lemma 3. *For all $\hat{y} \neq 0$ and all $r, u \in \bar{\mathcal{Y}} := \mathcal{Y} \cup \{\perp\}$: $\langle u|\mathsf{cO}_{x\hat{y}}|r\rangle = \gamma^{\hat{y}}_{u,r}$. Furthermore, $\mathsf{cO}_{x,\hat{0}} = \mathsf{I}$.*

The proof is a straightforward computation and is provided in the full version.

	$\perp$	$r \in \mathcal{Y}$
$\perp$	$\gamma^{\hat{y}}_{\perp,\perp} = 0$	$\gamma^{\hat{y}}_{\perp,r} = \dfrac{\hat{y}^*(r)}{\sqrt{M}}$
$u \in \mathcal{Y}$	$\dfrac{\hat{y}(u)}{\sqrt{M}}$	$\gamma^{\hat{y}}_{u,r} = \begin{cases} \left(1 - \dfrac{2}{M}\right)\hat{y}(u) + \dfrac{1}{M} & \text{if } u = r \in \mathcal{Y} \\ \dfrac{1 - \hat{y}(r) - \hat{y}(u)}{M} & \text{if } u \neq r \text{, both in } \mathcal{Y} \end{cases}$

Fig. 1. The evolution of the compressed oracle in the computational basis.

Since, for any fixed $\hat{y}$, the matrix $\mathsf{cO}_{x\hat{y}}$ is unitary, the squares of the absolute values of each column add to 1. Thus, for any $\hat{y}, r$ we can consider the probability distribution defined by $\tilde{P}[U = u|r, \hat{y}] := |\gamma^{\hat{y}}_{u,r}|^2$. This offers us a convenient notation, like $\tilde{P}[U \in \mathcal{S}|r, \hat{y}]$ for $\sum_{u \in \mathcal{S}} |\gamma^{\hat{y}}_{u,r}|^2$ or $\tilde{P}[U \neq r|r, \hat{y}]$ for $\sum_{u \neq r} |\gamma^{\hat{y}}_{u,r}|^2$. For later purposes, it is useful to observe that, for any $\mathsf{L} \subseteq \mathcal{Y}$ (i.e., $\perp \notin \mathsf{L}$),

$$\begin{aligned} \sum_r \tilde{P}[r \neq U \in \mathsf{L}|r, \hat{y}] &\leq \tilde{P}[U \in \mathsf{L}|\perp, \hat{y}] + \sum_{r \neq \perp} \tilde{P}[r \neq U \in \mathsf{L}|r, \hat{y}] \\ &\leq |\mathsf{L}|\frac{1}{M} + M|\mathsf{L}|\frac{9}{M^2} = 10P[U \in \mathsf{L}]\,, \end{aligned} \tag{8}$$

where $P[U \in \mathsf{L}] = \frac{|\mathsf{L}|}{M}$ is the probability for a random U in $\mathcal{Y}$ to be in L.

4.4 The Parallel-Query (Compressed) Oracle

Here, we extend the above compressed-oracle technique to the setting where a quantum algorithm may make *several* queries to the random oracle *in parallel*. Formally, for any positive integer k, a *k-parallel query* is given by k parallel applications of O, with the understanding that each application acts on a different input/output register pair. More explicitly, but slightly abusing notation of writing a k-th power, a k-parallel query is given by

$$\mathsf{O}^k : |\mathbf{x}\rangle|\mathbf{y}\rangle \otimes |H\rangle \mapsto |\mathbf{x}\rangle|\mathbf{y} + H(\mathbf{x})\rangle \otimes |H\rangle$$

for any $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$ and $\mathbf{y} = (y_1, \ldots, y_k) \in \mathcal{Y}^k$. The operator unitary $\mathsf{cO}^k := \mathsf{Comp} \circ \mathsf{O}^k \circ \mathsf{Comp}^\dagger$, which described the evolution of the compressed oracle under such a k-parallel query, then acts as

$$\mathsf{cO}^k : |\mathbf{x}\rangle|\hat{\mathbf{y}}\rangle \otimes |\varDelta\rangle \mapsto |\mathbf{x}\rangle|\hat{\mathbf{y}}\rangle \otimes \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}|\varDelta\rangle$$

for any $|\varDelta\rangle \in \mathbb{C}[\mathfrak{D}]$, where $\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}$ is the product $\mathsf{cO}_{x_1\hat{y}_1} \cdots \mathsf{cO}_{x_k\hat{y}_k}$. We recall that $\mathsf{cO}_{x_i\hat{y}_i}$ acts on register x_i (only), and $\mathsf{cO}_{x_i\hat{y}_i}$ and $\mathsf{cO}_{x_j\hat{y}_j}$ commute.

5 A Framework for Proving Quantum Query Bounds

In this section we set up a framework for proving lower-bounds on the query complexity (actually, equivalently, upper bounds on the success probability) of *quantum* algorithms in the (quantum) random oracle model. Our framework closely mimics the reasoning for classical algorithms and allows to easily "lift" the typical kind of reasoning to the quantum setting.

5.1 Setting Up the Framework

Definition 1. *A* database property *on $\mathfrak{D}$ is a subset $\mathsf{P} \subseteq \mathfrak{D}$.*

Remark 1. We think of P as a property that is either *true* or *false* for any $D \in \mathfrak{D}$. Furthermore, by convention, for any database property $\mathsf{P} \in \mathfrak{D}$, we overload notation and use P also to refer to the projection $\sum_{D \in \mathsf{P}} |D\rangle\!\langle D| \in \mathcal{L}(\mathbb{C}[\mathfrak{D}])$.

Three examples that we will later consider are

$$\begin{aligned}&\mathsf{PRMG} := \{D \,|\, \exists\, x : D(x) = 0\}\;, \quad \mathsf{CL} := \{D \,|\, \exists\, x, x' : D(x) = D(x') \neq \bot\} \quad \text{and}\\ &\mathsf{CHN}^q := \{D \,|\, \exists\, x_0, x_1, \ldots, x_q \in \mathcal{X} : D(x_{i-1}) \lhd x_i \;\forall i\}\;,\end{aligned}$$

where $\lhd$ denotes an arbitrary relation, e.g., $y \lhd x$ if y is a prefix of x.

We introduce the following notation. For any tuple $\mathbf{x} = (x_1, \ldots, x_k)$ of pairwise distinct $x_i \in \mathcal{X}$ and for any $D : \mathcal{X} \to \bar{\mathcal{Y}}$ we let

$$D|^{\mathbf{x}} := \left\{ D[\mathbf{x} \mapsto \mathbf{r}] \,|\, \mathbf{r} \in \bar{\mathcal{Y}}^k \right\} \subseteq \mathfrak{D}$$

be the set of databases that coincide with D outside of $\mathbf{x}$. Furthermore, for any database property $\mathsf{P} \subseteq \mathfrak{D}$, we then let

$$\mathsf{P}|_{D|^{\mathbf{x}}} := \mathsf{P} \cap D|^{\mathbf{x}}$$

be the restriction of P to the databases in $D|^{\mathbf{x}}$. We then typically think of $\mathsf{P}|_{D|^{\mathbf{x}}}$ as a property of functions $D' \in D|^{\mathbf{x}}$.

Remark 2. For fixed choices of $\mathbf{x}$ and D, we will often identify $D|^{\mathbf{x}}$ with $\bar{\mathcal{Y}}^k$ by means of the obvious map $\mathbf{r} \mapsto D[\mathbf{x} \mapsto \mathbf{r}]$. The property $\mathsf{P}|_{D|^{\mathbf{x}}}$ can then be considered to be a property/subset of $\bar{\mathcal{Y}}^k$, namely $\{\mathbf{r} \in \bar{\mathcal{Y}}^k \,|\, D[\mathbf{x} \mapsto \mathbf{r}] \in \mathsf{P}\}$. Accordingly, we do not distinguish between the projections

$$\sum_{D' \in \mathsf{P}|_{D|^{\mathbf{x}}}} |D'\rangle\!\langle D'| \in \mathcal{L}(\mathbb{C}[D|^{\mathbf{x}}]) \subseteq \mathcal{L}(\mathbb{C}[\mathfrak{D}]) \quad \text{and} \qquad \sum_{\substack{\mathbf{r} \in \bar{\mathcal{Y}}^k \\ D[\mathbf{x} \mapsto \mathbf{r}] \in \mathsf{P}}} |\mathbf{r}\rangle\!\langle \mathbf{r}| \in \mathcal{L}(\mathbb{C}[\bar{\mathcal{Y}}^k])$$

but refer to both as $\mathsf{P}|_{D|^{\mathbf{x}}}$, using our convention to use the same variable for a property and the corresponding projection. This is justified by the fact that on the space spanned by $|D[\mathbf{x} \mapsto \mathbf{r}]\rangle$ with $\mathbf{r} \in \bar{\mathcal{Y}}^k$, both act identically (with the understanding that the latter acts on the registers labeled by $\mathbf{x}$.). In particular, they have the same operator norm.

Example 1. For a given $\mathbf{x}$ and D, as a subset of $\bar{\mathcal{Y}}^k$, we have

$$\mathsf{PRMG}|_{D|^{\mathbf{x}}} = \begin{cases} \bar{\mathcal{Y}}^k & \text{if } D(\bar{x}) = 0 \text{ for some } \bar{x} \notin \{x_1, \ldots, x_k\} \\ \{\mathbf{r} \mid \exists\, i : r_i = 0\} & \text{else} \end{cases}$$

In words: if D has a zero outside of $\mathbf{x}$ then $D[\mathbf{x} \mapsto \mathbf{r}]$ has a zero for any $\mathbf{r} \in \bar{\mathcal{Y}}^k$; otherwise, $D[\mathbf{x} \mapsto \mathbf{r}]$ has a zero if and only if one of the coordinates of $\mathbf{r}$ is zero.

The following definition is the first main ingredient of our framework. The upcoming theorem, which relates the success probability of a quantum algorithm to the quantum transition capacity, then forms the second main ingredient.

Definition 2 (Quantum transition capacity). *Let P, P' be two database properties. Then, the* quantum transition capacity *(of order k) is defined as*

$$\llbracket \mathsf{P} \xrightarrow{k} \mathsf{P}' \rrbracket := \max_{\mathbf{x}, \hat{\mathbf{y}}, D} \| \mathsf{P}'|_{D|^{\mathbf{x}}} \, \mathsf{cO}_{\mathbf{x}\mathbf{y}} \, \mathsf{P}|_{D|^{\mathbf{x}}} \| \, .$$

Furthermore, we define

$$\llbracket \mathsf{P} \overset{k,q}{\Longrightarrow} \mathsf{P}' \rrbracket := \sup_{U_2, \ldots, U_q} \| \mathsf{P}' \mathsf{cO}^k \, U_q \, \mathsf{cO} \cdots \mathsf{cO}^k \, U_2 \, \mathsf{cO}^k \, \mathsf{P} \| \, .$$

where the supremum is over all positive $d \in \mathbb{Z}$ and all unitaries $U_2, \ldots, U_q$ acting on $\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{Y}] \otimes \mathbb{C}^d$.

By definition, the notion $\llbracket \mathsf{P} \overset{k,q}{\Longrightarrow} \mathsf{P}' \rrbracket$ equals the square-root of the maximal probability that the internal state of the compressed oracle, when supported by databases $D \in \mathsf{P}$, turns into a database $D' \in \mathsf{P}'$ by means of a quantum query algorithm that performs q k-parallel queries, and when we then measure the internal state. In particular, for p' as in Corollary 1 and P^R as below in Theorem 1, it holds that $\llbracket \bot \overset{k,q}{\Longrightarrow} \mathsf{P}^R \rrbracket = \sqrt{p'}$.

Similarly, but on a more intuitive level so far, $\llbracket \mathsf{P} \xrightarrow{k} \mathsf{P}' \rrbracket$ represents a measure of how likely it is that, as a result of *one* k-parallel query, a database $D \in \mathfrak{D}$ that satisfies P turns into a database D' that satisfies P'. In the context of these two notations, $\bot$ is understood to be the database property that is satisfied by $\bot \in \mathfrak{D}$ only, and $\neg\mathsf{P}$ is the complement of P, i.e., $\neg\mathsf{P} = \mathsf{I} - \mathsf{P}$ (as projections). We also write $\mathsf{P} \to \mathsf{P}'$ and refer to this as a *database transition* when considering two database properties P and P' in the context of the above two notions. Formally, they are related as follows.

Lemma 4. *For any sequence of database properties $\mathsf{P}_0, \mathsf{P}_1, \ldots, \mathsf{P}_q$,*

$$\llbracket \neg\mathsf{P}_0 \overset{k,q}{\Longrightarrow} \mathsf{P}_q \rrbracket \le \sum_{s=1}^{q} \llbracket \neg\mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s \rrbracket \, .$$

Proof. By means of inserting $\mathsf{I} = \mathsf{P}_q + (\mathsf{I} - \mathsf{P}_q)$ before U_q and using properties of the norm, we obtain

$$\|\mathsf{P}_q \, \mathsf{cO}^k \, U_q \, \mathsf{cO}^k \cdots \mathsf{cO}^k \, (\mathsf{I} - \mathsf{P}_0)\| \le \|\mathsf{P}_{q-1} \, \mathsf{cO}^k \cdots \mathsf{cO}^k \, (\mathsf{I} - \mathsf{P}_0)\| + \|\mathsf{P}_q \, \mathsf{cO}^k \, U_q \, (\mathsf{I} - \mathsf{P}_{q-1})\| \, .$$

To the first term, we apply induction; so it remains to bound the second term by $\llbracket \neg\mathsf{P}_{q-1} \xrightarrow{k} \mathsf{P}_q \rrbracket$. Using that U_q and P_{q-1} commute (as they act on different subsystems) and setting $\mathsf{P} = \neg\mathsf{P}_{q-1}$ and $\mathsf{P}' = \mathsf{P}_q$, this follows from[8]

$$\|\mathsf{P}'\mathsf{cO}^k\,\mathsf{P}\| \le \max_{\mathbf{x},\hat{\mathbf{y}}} \|\mathsf{P}'\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\,\mathsf{P}\| \le \max_{\mathbf{x},\hat{\mathbf{y}},D} \|\mathsf{P}'|_{D|^{\mathbf{x}}}\,\mathsf{cO}_{\mathbf{x}\mathbf{y}}\,(\mathsf{I}-\mathsf{P}|_{D|^{\mathbf{x}}})\|\,,$$

where for the first inequality we observe that $\mathsf{P}'\mathsf{cO}^k\mathsf{P}$ maps $|\mathbf{x}\rangle|\hat{\mathbf{y}}\rangle \otimes |\Gamma\rangle$ to $|\mathbf{x}\rangle|\hat{\mathbf{y}}\rangle \otimes \mathsf{P}'\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\mathsf{P}|\Gamma\rangle$, and so the first inequality holds by basic properties of the operator norm. Similarly for the second inequality: For any fixed D, consider the subspace of $\mathbb{C}[\mathfrak{D}]$ spanned by $|D[\mathbf{x}\mapsto\mathbf{r}]\rangle$ with $\mathbf{r} \in \bar{\mathcal{Y}}^k$. On this subspace, P and $\mathsf{P}|_{D|^{\mathbf{x}}}$ are identical projections (and similarly for P'). Also, $\mathsf{cO}_{\mathbf{x}\mathbf{y}}$ is a unitary on this subspace. The claim then again follows again by basic properties of the operator norm.

The following is now a direct consequence of Corollary 1, the definition of $\llbracket \bot \overset{k,q}{\Longrightarrow} \mathsf{P}^R \rrbracket$, and the above lemma.

Theorem 1. *Let R be a relation, and let $\mathcal{A}$ be a k-parallel q-query quantum oracle algorithm with success probability p, as considered in Corollary 1. Consider the database property induced by R, given as*

$$\mathsf{P}^R = \bigl\{ D \in \mathfrak{D} \,|\, \exists\, \mathbf{x} \in \mathcal{X}^\ell : \bigl(\mathbf{x}, D(\mathbf{x})\bigr) \in R \bigr\}\,.$$

Then, for any database properties $\mathsf{P}_0, \ldots, \mathsf{P}_q$ with $\mathsf{P}_0 = \neg\bot$ and $\mathsf{P}_q = \mathsf{P}^R$:

$$\sqrt{p} \,\le\, \llbracket \bot \overset{k,q}{\Longrightarrow} \mathsf{P}^R \rrbracket + \sqrt{\frac{\ell}{M}} \,\le\, \sum_{s=1}^{q} \llbracket \neg\mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s \rrbracket + \sqrt{\frac{\ell}{M}}\,.$$

Remark 3. This result implies that in order to bound p, it is sufficient to find a sequence $\bot \notin \mathsf{P}_0, \ldots, \mathsf{P}_q = \mathsf{P}^R$ of properties for which all quantum transition capacities $\llbracket \neg\mathsf{P}_{s-1} \to \mathsf{P}_s \rrbracket$ are small. Often, it is good to keep track of the (growing but bounded) size of the database and instead bound the capacities

$$\llbracket \mathsf{SZ}_{\le k(s-1)} \backslash \mathsf{P}_{s-1} \to \mathsf{P}_s \rrbracket = \llbracket \mathsf{SZ}_{\le k(s-1)} \backslash \mathsf{P}_{s-1} \to \mathsf{P}_s \cup \neg\mathsf{SZ}_{\le ks} \rrbracket\,,$$

where the equality is due to the fact that the size of a database cannot grow by more than k with one k-parallel query. Formally, we would then consider the database properties $\mathsf{P}'_s = \neg(\mathsf{SZ}_{\le ks} \setminus \mathsf{P}_s) = \mathsf{P}_s \cup \neg\mathsf{SZ}_{\le ks}$.

In the following section, we offer techniques to bound the quantum transition capacities (in certain cases) using *purely classical* reasoning. In connection with Theorem 1, this then provides means to prove lower bounds on the quantum query complexity (for certain computational problems in the random oracle model) using purely classical reasoning.

[8] In line with Remark 2, we consider $\mathsf{P}|_{D|^{\mathbf{x}}}$ to be a projection acting on $\mathbb{C}[\bar{\mathcal{Y}}^k]$, and thus I in the last term is the identity in $\mathcal{L}(\mathbb{C}[\bar{\mathcal{Y}}^k])$.

5.2 Bounding Transition Capacities Using Classical Reasoning Only

The idea is to "recognize" a database transition $\neg\mathsf{P} \to \mathsf{P}$ in terms of *local* properties L, for which the truth value of $D \in^? \mathsf{L}$, i.e. whether $D \in \mathsf{L}$ or not, only depends on the function value $D(x)$ at *one single point* x (or at few points), and to use that the behavior of the compressed oracle at a single point x is explicitly given by Lemma 3. In the following two sections, we consider two possible ways to do this, but first, we provide the formal definition for local properties.

Definition 3. *A database property* $\mathsf{L} \subseteq \mathfrak{D}$ *is* ℓ-local *if* $\exists\, \mathbf{x} = (x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$ *so that*

1. *the truth value of* $D \in^? \mathsf{L}$ *is uniquely determined by* $D(\mathbf{x})$*, and*
2. *if* $D \in \mathsf{L} \wedge (\exists\, i \in \{1, \ldots, \ell\} : D(x_i) = \bot)$ *then* $D[x_i \mapsto r_i] \in \mathsf{L}\ \forall\, r_i \in \mathcal{Y}$*.*

The set $\{x_1, \ldots, x_\ell\}$ *is then called the* support *of* L*, and denoted by* $\mathsf{Supp}(\mathsf{L})$*.*

Remark 4. We observe that, as defined above, the support of an ℓ-local property is not necessarily uniquely defined: if ℓ is not minimal with the required property then there are different choices. A natural way to have a unique definition for $\mathsf{Supp}(\mathsf{L})$ is to require it to have minimal size. For us, it will be more convenient to instead consider the choice of the support to be part of the specification of L. Furthermore, we then declare that $\mathsf{Supp}(\mathsf{L} \cup \mathsf{M}) = \mathsf{Supp}(L) \cup \mathsf{Supp}(\mathsf{M})$, and $\mathsf{Supp}(\mathsf{L}|_{D|^{\mathbf{x}}}) = \mathsf{Supp}(\mathsf{L}) \cap \{x_1, \ldots, x_k\}$ for any $D \in \mathfrak{D}$ and $\mathbf{x} = (x_1, \ldots, x_k)$.

Remark 5. Condition 2 captures that $\bot$ is a special dummy symbol with no more "value" than any other $r \in \mathcal{Y}$.

For example, for any database property P, any $\mathbf{x} = (x_1, \ldots, x_\ell)$ and D, the property $\mathsf{P}|_{D|^{\mathbf{x}}}$ satisfies requirement 1. of Definition 3. In line with this, Remark 2 applies here as well: we may identify an ℓ-local property L with a subset of $\bar{\mathcal{Y}}^\ell$.

Reasoning via Strong Recognizability

Definition 4. *A database transition* $\neg\mathsf{P} \to \mathsf{P}'$ *is* (uniformly) strongly recognizable *by* ℓ*-local properties if there exists a family of* ℓ*-local properties* $\{\mathsf{L}_i\}_i$ *with*

$$\mathsf{P}' \subseteq \bigcup_i \mathsf{L}_i \subseteq \mathsf{P}\,. \tag{9}$$

We also consider the following weaker but somewhat more intricate version.

Definition 5. *A database transition* $\neg\mathsf{P} \to \mathsf{P}'$ *is said be* k-non-uniformly strongly recognizable *by* ℓ*-local properties if for every* $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$ *with disjoint entries, and for every* $D \in \mathfrak{D}$*, there exist a family* $\{\mathsf{L}_i^{\mathbf{x},D}\}_i$ *of* ℓ*-local properties* $\mathsf{L}_i^{\mathbf{x},D}$ *with supports in* $\{x_1, \ldots, x_k\}$ *so that*

$$\mathsf{P}'|_{D|^{\mathbf{x}}} \subseteq \bigcup_i \mathsf{L}_i^{\mathbf{x},D} \subseteq \mathsf{P}|_{D|^{\mathbf{x}}}\,. \tag{10}$$

It is easiest to think about these definitions for the case $\mathsf{P} = \mathsf{P}'$, where (9) and (10) become equalities. Requirement (9) then means that for D to satisfy P it is *necessary* and *sufficient* that D satisfies one of the local properties.

Remark 6. In the above definitions, as long as the support-size remains bounded by ℓ, one can always replace two properties by their union without affecting (9), respectively (10). Thus, we may — and by default do — assume the L_i's to have *distinct* (though not necessarily disjoint) supports in Definition 4, and the same we may assume for the $\mathsf{L}_i^{\mathbf{x},D}$'s for every $\mathbf{x}$ and D in Definition 5.

Remark 7. Definition 4 implies Definition 5 with $\mathsf{L}_i^{\mathbf{x},D} := \mathsf{L}_i|_{D|^{\mathbf{x}}}$.

Theorem 2. *Let $\neg\mathsf{P} \to \mathsf{P}'$ be k-non-uniformly strongly recognizable by 1-local properties $\{\mathsf{L}_1^{\mathbf{x},D}, \ldots, \mathsf{L}_k^{\mathbf{x},D}\}$, where, w.l.o.g., the support of $\mathsf{L}_i^{\mathbf{x},D}$ is $\{x_i\}$. Then*

$$\llbracket \neg\mathsf{P} \xrightarrow{k} \mathsf{P}' \rrbracket \leq \max_{\mathbf{x},D} \sqrt{10 \sum_i P\big[U \in \mathsf{L}_i^{\mathbf{x},D}\big]}$$

with the convention that $P\big[U \in \mathsf{L}_i^{\mathbf{x},D}\big] = 0$ if $\mathsf{L}_i^{\mathbf{x},D}$ is constant true or false.

Before doing the proof, let us look at one of the considered examples.

Example 2. $\mathsf{P}' = \mathsf{P} = \mathsf{PRMG}$ is uniformly strongly recognized by the 1-local properties $\mathsf{L}_x = \{D | D(x) = 0\}$. Also, as a subset of $\bar{\mathcal{Y}}$, the property $\mathsf{L}_x^{\mathbf{x},D} := \mathsf{L}_x|_{D|^{\mathbf{x}}}$ is either $\{0\}$ or constant true or false.[9] In the non-constant case, we obviously have $P\big[U \in \mathsf{L}_i^{\mathbf{x},D}\big] = P[U = 0] = 1/M$. It then follows from Theorem 2 that we can bound the transition capacity as $\llbracket \neg\mathsf{PRMG} \xrightarrow{k} \mathsf{PRMG} \rrbracket \leq \sqrt{10k/M}$ and thus from Theorem 1, setting $\mathsf{P}_i = \mathsf{PRMG}$ for all i, that the probability p of any k-parallel q-query algorithm outputting a 0-preimage x is bounded by

$$p \leq \Big(q\sqrt{\tfrac{10k}{M}} + \tfrac{1}{\sqrt{M}} \Big)^2 = O\big(\tfrac{kq^2}{M}\big) \,.$$

Proof (of Theorem 2). Consider arbitrary $\mathbf{x}$ and D. To simplify notation, we then write L_i for $\mathsf{L}_i^{\mathbf{x},D}$. We introduce the properties $\mathsf{M}_i := \mathsf{L}_i \setminus (\bigcup_{j<i} \mathsf{L}_j)$ for $1 \leq i \leq k$. By assumption (10), as projectors they satisfy the operator inequalities $\mathsf{P}'|_{D|^{\mathbf{x}}} \leq \sum_i \mathsf{M}_i \leq \sum_i \mathsf{L}_i$ and $\mathsf{M}_i \leq \mathsf{L}_i \leq \mathsf{P}|_{D|^{\mathbf{x}}}$ $\forall i$, where, on top, the M_i's are mutually orthogonal. Then, exploiting the various properties, for any $\hat{\mathbf{y}}$ we have

$$\|\mathsf{P}'|_{D,\mathbf{x}}\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, (\mathsf{I} - \mathsf{P}|_{D,\mathbf{x}})\|^2 \leq \Big\| \sum_i \mathsf{M}_i\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, (\mathsf{I} - \mathsf{P}|_{D|^{\mathbf{x}}}) \Big\|^2$$

$$= \sum_i \|\mathsf{M}_i\, \mathsf{cO}_{\hat{\mathbf{y}}}\, (\mathsf{I} - P|_{D|^{\mathbf{x}}})\|^2 \leq \sum_i \|\mathsf{L}_i\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, (\mathsf{I} - \mathsf{L}_i)\|^2 = \sum_i \|\mathsf{L}_i\, \mathsf{cO}_{x_i \hat{y}_i}\, (\mathsf{I} - \mathsf{L}_i)\|^2 \,,$$

[9] In more detail, $\mathsf{L}_x|_{D|^{\mathbf{x}}} = \{0\}$ whenever $x \in \{x_1, \ldots, x_k\}$, and otherwise it is constant true if $D(x) = 0$ and constant false if $D(x) \neq 0$.

where, by considering the map as a map on $\mathbb{C}[\bar{\mathcal{Y}}]$ and bounding the operator norm by the Frobenius norm,

$$\begin{aligned}\|\mathsf{L}_i\,\mathsf{cO}_{x_i\hat{y}_i}\,(\mathsf{I}-\mathsf{L}_i)\|^2 &\le \sum_{r_i,u_i\in\bar{\mathcal{Y}}} |\langle u_i|\mathsf{L}_i\,\mathsf{cO}_{x_i\hat{y}_i}\,(\mathsf{I}-\mathsf{L}_i)|r_i\rangle|^2 \\ &= \sum_{\substack{r_i\not\in\mathsf{L}_i\\ u_i\in\mathsf{L}_i}} |\langle u_i|\mathsf{cO}_{x_i\hat{y}_i}|r_i\rangle|^2 = \sum_{r_i\not\in\mathsf{L}_i} \tilde{P}[U\in\mathsf{L}_i|r_i,\hat{y}_i]\,.\end{aligned}$$

The claim now follows from (8), with the additional observations that if $\bot\in\mathsf{L}_i$ (in which case (8) does not apply) then L_i is constant-true (by property 2 of Definition 3), and that the sum is empty if L_i is constant-true. □

Reasoning via Weak Recognizability Here, we consider a weaker notion of recognizability, which is wider applicable but results in a slightly worse bound. Note that it will be more natural here to speak of a transition $\mathsf{P}\to\mathsf{P}'$ instead of $\neg\mathsf{P}\to\mathsf{P}'$, i.e., we now write P for what previously was its complement.

Definition 6. *A database transition* $\mathsf{P}\to\mathsf{P}'$ *is* (uniformly) weakly recognizable *by ℓ-local properties if there exists a family of ℓ-local properties $\{\mathsf{L}_i\}_i$ so that*

$$D\in\mathsf{P}\;\wedge\;D'\in\mathsf{P}'\implies\exists\,i:D'\in\mathsf{L}_i\;\wedge\;\big(\exists\,x\in\mathsf{Supp}(\mathsf{L}_i):D(x)\neq D'(x)\big)\,.$$

Also here, we have a non-uniform version (see below). Furthermore, Remarks 6 and 7 apply correspondingly; in particular, we may assume the supports in the considered families of local properties to be distinct.

Definition 7. *A database transition* $\mathsf{P}\to\mathsf{P}'$ *is said be* k-non-uniformly weakly recognizable *by ℓ-local properties if for every $\mathbf{x}=(x_1,\dots,x_k)\in\mathcal{X}^k$ with disjoint entries, and for every $D\in\mathfrak{D}$, there exist a family of ℓ-local properties $\{\mathsf{L}_i^{\mathbf{x},D}\}_i$ with supports in $\{x_1,\dots,x_k\}$ so that*

$$\begin{aligned}&D_\circ\in\mathsf{P}|_{D|^{\mathbf{x}}}\;\wedge\;D'\in\mathsf{P}'|_{D|^{\mathbf{x}}}\\ &\qquad\implies\exists\,i:D'\in\mathsf{L}_i^{\mathbf{x},D}\;\wedge\;\big(\exists\,x\in\mathsf{Supp}(\mathsf{L}_i^{\mathbf{x},D}):D_\circ(x)\neq D'(x)\big)\,.\end{aligned}\tag{11}$$

Remark 8. Viewing $\mathsf{L}_i^{\mathbf{x},D}$ as subset of $\bar{\mathcal{Y}}^k$, and its support $\mathsf{L}_i^{\mathbf{x},D}=\{x_{i_1},\dots,x_{i_\ell}\}$ then as subset $\{i_1,\dots,i_\ell\}$ of $\{1,\dots,k\}$, (11) can equivalently be written as follows, which is in line with Lemma 1 (where $\mathsf{Supp}(\mathsf{L}_i^{\mathbf{x},D})=\{i\}$):

$$D[\mathbf{x}\mapsto\mathbf{r}]\in\mathsf{P}\wedge D[\mathbf{x}\mapsto\mathbf{u}]\in\mathsf{P}'\implies\exists\,i:\mathbf{u}\in\mathsf{L}_i^{\mathbf{x},D}\wedge\big(\exists\,j\in\mathsf{Supp}(\mathsf{L}_i^{\mathbf{x},D}):\mathbf{r}_j\neq\mathbf{u}_j\big).$$

Example 3. Consider $\mathsf{CHN}^q=\{D\,|\,\exists\,x_0,x_1,\dots,x_q\in\mathcal{X}:D(x_{i-1})\triangleleft x_i\;\forall i\}$ for an arbitrary positive integer q. For any $\mathbf{x}$ and D, we let $\mathsf{L}_i=\mathsf{L}_i^{\mathbf{x},D}$ be the 1-local property that has support $\{x_i\}$ and, as a subset of $\bar{\mathcal{Y}}$, is defined as (4), i.e., so that $u\in\mathsf{L}_i$ if and only if $u\triangleleft x$ for some x with $D(x)\neq\bot$ or $x\in\{x_1,\dots,x_k\}$. Lemma 1 from the classical analysis shows that condition (11) is satisfied for the database transition $\neg\mathsf{CHN}^q\to\mathsf{CHN}^{q+1}$. This in particular implies that (11) is satisfied for the database transition $\mathsf{SZ}_{\le k(q-1)}\setminus\mathsf{CHN}^q\to\mathsf{CHN}^{q+1}$.

Theorem 3. *Let $\mathsf{P} \to \mathsf{P}'$ be k-non-uniformly weakly recognizable by 1-local properties $\mathsf{L}_i^{\mathbf{x},D}$, where the support of $\mathsf{L}_i^{\mathbf{x},D}$ is $\{x_i\}$ or empty. Then*

$$[\![\mathsf{P} \xrightarrow{k} \mathsf{P}']\!] \le \max_{\mathbf{x},D} e \sum_i \sqrt{10P\big[U \in \mathsf{L}_i^{\mathbf{x},D}\big]}\,,$$

where e is Euler's number.

Example 4. In the above example regarding CHN^q with the considered $\mathsf{L}_i^{\mathbf{x},D}$'s for $D \in \mathsf{SZ}_{\le kq}$, as in the derivation of the classical bound in Section 2.2, it holds that $P[U \in \mathsf{L}_i^{\mathbf{x},D}] \le kqT/M$, where T denotes the maximal number of $y \in \mathcal{Y}$ with $y \lhd x$ (for any x). For $D \notin \mathsf{SZ}_{\le kq}$ we may then choose $\mathsf{L}_i^{\mathbf{x},D} := \emptyset$. Thus,

$$[\![\mathsf{SZ}_{\le k(q-1)} \backslash \mathsf{CHN}^q \xrightarrow{k} \mathsf{CHN}^{q+1}]\!] \le ek\sqrt{\frac{10kqT}{M}}\,,$$

and applying Theorem 1 (and the subsequent remark) to the database transitions $\mathsf{SZ}_{\le k(s-1)} \setminus \mathsf{CHN}^s \to \mathsf{CHN}^{s+1}$ for $s = 1, \ldots, q$, we obtain the following bound, which we state as a theorem here given that this is a new bound.

Theorem 4. *Let $\lhd$ be a relation over $\mathcal{Y}$ and $\mathcal{X}$. The probability p of any k-parallel q-query oracle algorithm $\mathcal{A}$ outputting $x_0, x_1, \ldots, x_{q+1} \in \mathcal{X}$ with the property that $H(x_i) \lhd x_{i+1}$ for all $i \in \{0, \ldots, q\}$ is bounded by*

$$p \le \left(qk\sqrt{\frac{10qkT}{M}}e + \sqrt{\frac{q+2}{M}}\right)^2 = O\left(\frac{q^3k^3T}{M}\right),$$

where $T := \max_x |\{y \in \mathcal{Y} \,|\, y \lhd x\}|$ and $M := |\mathcal{Y}|$.

Proof (of Theorem 3). We consider fixed choices of $\mathbf{x}$ and D, and we then write L_i for $\mathsf{L}_i^{\mathbf{x},D}$. For arbitrary but fixed $\hat{\mathbf{y}}$, we introduce

$$A_i := \sum_{\substack{u_i, r_i \text{ s.t.} \\ u_i \in \mathsf{L}_i \wedge r_i \neq u_i}} |u_i\rangle\langle u_i|\, \mathsf{cO}_{x_i\hat{y}_i} |r_i\rangle\langle r_i| \qquad \text{and}$$

$$B_i := \mathsf{cO}_{x_i\hat{y}_i} - A_i = \sum_{\substack{u_i, r_i \text{ s.t.} \\ u_i \notin \mathsf{L}_i \vee r_i = u_i}} |u_i\rangle\langle u_i|\, \mathsf{cO}_{x_iy_i} |r_i\rangle\langle r_i|$$

and observe that, taking it as understood that the operators $\mathsf{cO}_{x_1\hat{y}_1}, \ldots, \mathsf{cO}_{x_k\hat{y}_k}$ act on different subsystems,

$$\begin{aligned}
\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}} = \prod_{j=1}^{k} \mathsf{cO}_{x_j\hat{y}_j} &= \prod_{j=1}^{k-1} \mathsf{cO}_{x_j\hat{y}_j} A_k + \prod_{j=1}^{k-1} \mathsf{cO}_{x_j\hat{y}_j} B_k \\
&= \prod_{j=1}^{k-1} \mathsf{cO}_{x_j\hat{y}_j} A_k + \prod_{j=1}^{k-2} \mathsf{cO}_{x_j\hat{y}_j} A_{k-1} B_k + \prod_{j=1}^{k-2} \mathsf{cO}_{x_j\hat{y}_j} B_{k-1} B_k \\
&= \cdots = \sum_{i=0}^{k} \Big(\prod_{j<k-i} \mathsf{cO}_{x_j\hat{y}_j} \Big) A_{k-i} \Big(\prod_{j>k-i} B_j \Big)
\end{aligned}$$

with the convention that $A_0 = \mathsf{I}$. Furthermore, by assumption on the L_i's, it follows that

$$Q := \mathsf{P}'|_{D|^{\mathbf{x}}} \Bigl(\prod_{j>0} B_j \Bigr) \mathsf{P}|_{D|^{\mathbf{x}}} = 0 \,.$$

Indeed, by definition of $\mathsf{P}'|_{D|^{\mathbf{x}}}$ and $\mathsf{P}|_{D|^{\mathbf{x}}}$ (considering them as subsets of $\bar{\mathcal{Y}}^k$ now), for $\langle \mathbf{u}|Q|\mathbf{r}\rangle$ *not* to vanish, it is necessary that $\mathbf{r} \in \mathsf{P}|_{D|^{\mathbf{x}}}$ and $\mathbf{u} \in \mathsf{P}'|_{D|^{\mathbf{x}}}$. But then, by assumption, for such $\mathbf{r}$ and $\mathbf{u}$ there exists i so that $u_i \in \mathsf{L}_i$ and $r_i \neq u_i$, and thus for which $\langle u_i|B_i|r_i\rangle = 0$. Therefore, $\langle \mathbf{u}|Q|\mathbf{r}\rangle = \langle \mathbf{u}|\prod_j B_j|\mathbf{r}\rangle = \prod_j \langle u_j|B_j|r_j\rangle$ still vanishes. As a consequence, we obtain

$$\begin{aligned} \|\mathsf{P}'|_{D|^{\mathbf{x}}}\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, \mathsf{P}|_{D|^{\mathbf{x}}}\| &\leq \Biggl\| \sum_{i=0}^{k-1} \Bigl(\prod_{j<k-i} \mathsf{cO}_{x_j \hat{y}_j} \Bigr) A_{k-i} \Bigl(\prod_{j>k-i} B_j \Bigr) \Biggr\| \\ &\leq \sum_{i=0}^{k-1} \Bigl(\|A_{k-i}\| \prod_{j>k-i} \|B_j\| \Bigr) . \end{aligned}$$

Using that $\|B_i\| = \|\mathsf{cO}_{x_i \hat{y}_i} - A_i\| \leq 1 + \|A_i\|$, this is bounded by

$$\leq \sum_{i=1}^{k} \|A_i\| \prod_{j=1}^{k} (1 + \|A_j\|) \leq \sum_i \|A_i\|\, e^{\sum_j \ln(1+\|A_j\|)} \leq \sum_i \|A_i\|\, e$$

where the last inequality uses that $\ln(1 + \|A_j\| \leq \|A_j\|$, and so the last inequality holds if $\sum_j \|A_j\| \leq 1$, while the final term is trivially an upper bound on the figure of merit otherwise. Using the fact that the operator norm is upper bounded by the Frobenius norm, we observe that

$$\|A_i\|^2 \leq \sum_{r_i, u_i} |\langle u_i|A_i|r_i\rangle|^2 = \sum_{\substack{u_i, r_i \text{ s.t.} \\ u_i \in L_i \wedge r_i \neq u_i}} |\langle u_i|\mathsf{cO}_{x_i y_i}|r_i\rangle|^2 = \sum_{r_i} \tilde{P}[r_i \neq U \in L_i | r_i, y_i] \,,$$

and the final term is bounded by $10P[U \in \mathsf{L}_i]$ due to (8), here with the additional observation that if $\perp \in \mathsf{L}_i$ (and so (8) does not apply) then, by condition 2 of Definition 3, $\mathsf{L}_i = \bar{\mathcal{Y}}$, and hence the bound holds trivially. □

General ℓ-Locality and Collision Finding We now remove the limitation on the locality being $\ell = 1$. The bound then becomes a bit more intricate, and we only have a version for *strong* recognizability.

Theorem 5. *Let $\mathsf{P} \to \mathsf{P}'$ be a database transition that is k-non-uniformly strongly recognizable by ℓ-local properties L_t, where we leave the dependency of $\mathsf{L}_t = \mathsf{L}_t^{\mathbf{x},D}$ on $\mathbf{x}$ and D implicit. Then*

$$[\![\mathsf{P} \xrightarrow{k} \mathsf{P}']\!] \leq \max_{\mathbf{x},D} e\ell \sqrt{10 \sum_t \max_{x \in \mathsf{Supp}(\mathsf{L}_t)} \max_{D' \in D|^{\mathsf{Supp}(\mathsf{L}_t)}} P\bigl[U \in \mathsf{L}_t|_{D'|^x}\bigr]} \,.$$

with the convention that $P\bigl[U \in \mathsf{L}_t|_{D'|^x}\bigr]$ vanishes if $\mathsf{L}_t|_{D'|^x}$ is trivial.

The proof is given in the full version [9]; it combines techniques from the proofs of Theorem 2 and Theorem 3.

Example 5. Consider $\mathsf{CL} = \{D \,|\, \exists\, x, x' : D(x) = D(x') \neq \bot\}$. For any $D \in \mathfrak{D}$ and $\mathbf{x} = (x_1, \dots, x_k)$, consider the family of 2-local properties consisting of

$$\mathsf{CL}_{i,j} := \{D_\circ \in D|^{\mathbf{x}} \,|\, D_\circ(x_i) = D_\circ(x_j) \neq \bot\} \qquad \text{and}$$
$$\mathsf{CL}_i := \{D_\circ \in D|^{\mathbf{x}} \,|\, \exists\, \bar{x} \notin \{x_1, \dots, x_k\} : D_\circ(x_i) = D(\bar{x}) \neq \bot\}$$

for $i \neq j \in \{1, \dots, k\}$, with respective supports $\{x_i, x_j\}$ and $\{x_i\}$.

It is easy to see that this family of 2-local properties satisfies (10) for the database transition $\neg\mathsf{CL} \to \mathsf{CL}$. Indeed, if D and D' are identical outside of $\mathbf{x}$, and D has no collision while D' has one, then D''s collision must be for x_i, x_j inside $\mathbf{x}$, or for one x_i inside and one $\bar{x}$ outside. As an immediate consequence, the family also satisfies (10) for the database transition $(\mathsf{SZ}_{\leq ks} \setminus \mathsf{CL}) \to \mathsf{CL}$. In this case though, whenever $D \notin \mathsf{SZ}_{\leq k(s+1)}$ the left hand side of (10) is never satisfied and so we may replace the family of local properties to consist of (only) the constant-false property.

Consider $\mathbf{x} = (x_1, \dots, x_k)$ and $D \in \mathsf{SZ}_{\leq k(s+1)}$ with $s \leq q$. Then, for $i \neq j$, as subsets of $\bar{\mathcal{Y}}$ we have that

$$\mathsf{CL}_{i,j}|_{D'|^{x_i}} = \{D'(x_j)\} \quad \text{and} \quad \mathsf{CL}_i|_{D'|^{x_i}} = \{D'(\bar{x}) \,|\, \bar{x} \notin \{x_1, \dots, x_k\} : D'(\bar{x}) \neq \bot\}$$

for any $D' \in D|^{(x_i, x_j)}$ and $D' \in D|^{x_i}$, respectively, and therefore

$$P\big[U \in \mathsf{CL}_{i,j}|_{D'|^{x_i}}\big] = \frac{1}{M} \qquad \text{and} \qquad P\big[U \in \mathsf{CL}_i|_{D'|^{x_i}}\big] \leq \frac{kq}{M}\,.$$

So, by Theorem 5,

$$[\![\mathsf{SZ}_{\leq ks} \backslash \mathsf{CL} \xrightarrow{k} \mathsf{CL}]\!] \leq 2e\sqrt{10\big(\tfrac{k^2}{M} + \tfrac{k^2 q}{M}\big)} = 2ek\sqrt{10\,\tfrac{q+1}{M}}$$

and hence, by Theorem 1, we obtain the following bound.

Theorem 6. *The probability p of any k-parallel q-query algorithm outputting a collision is bounded by*

$$p \leq \left(2qek\sqrt{10\,\frac{q+1}{M}} + \frac{2}{\sqrt{M}}\right)^2 = O\!\left(\frac{k^2 q^3}{M}\right).$$

5.3 Some Rules for the Quantum Transition Capacity

As we have seen, certain "simple" lower bounds on the query complexity (respectively upper bounds on the success probability) can be obtained rather directly by bounding the quantum transition capacity by the means discussed above. In more complex scenarios, as we will encounter in the next section, it will be convenient to first *manipulate* the quantum transition capacity, e.g., to decompose it into different cases that can then be analyzed individually. We thus collect some useful manipulation rules here; all the proofs can be found in the full version [9].

To start with, since $\mathsf{cO}^\dagger_{\mathbf{x}\hat{\mathbf{y}}} = \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}^*}$, we note that the quantum transition capacity is symmetric:

$$\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket = \llbracket \mathsf{P}' \overset{k}{\rightarrow} \mathsf{P} \rrbracket \, .$$

Therefore, the following bounds hold correspondingly also for $\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \cap \mathsf{Q} \rrbracket$ etc.

Lemma 5. *For any database properties* P, P' *and* Q*,*

$$\llbracket \mathsf{P} \cap \mathsf{Q} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket \; \leq \min\bigl\{ \llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket, \llbracket \mathsf{Q} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket \bigr\} \qquad \textit{and}$$

$$\max\bigl\{ \llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket, \llbracket \mathsf{Q} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket \bigr\} \leq \llbracket \mathsf{P} \cup \mathsf{Q} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket \; \leq \llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket + \llbracket \mathsf{Q} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket \, .$$

Corollary 2. *If* $\mathsf{P} \subseteq \mathsf{Q}$ *then* $\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket \leq \llbracket \mathsf{Q} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket$ *and* $\llbracket \mathsf{P}' \overset{k}{\rightarrow} \mathsf{P} \rrbracket \leq \llbracket \mathsf{P}' \overset{k}{\rightarrow} \mathsf{Q} \rrbracket$.

In the following, we extend the definition of the quantum transition capacity as follows, which captures a restriction of the query vector $\mathbf{x} = (x_1, \ldots, x_k)$ to entries x_i in $X \subseteq \mathcal{X}$.

$$\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \big| X \rrbracket := \max_{\substack{\mathbf{x} \in X^k \\ \hat{\mathbf{y}}, D}} \| \mathsf{P}'|_{D|^{\mathbf{x}}} \, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}} \, \mathsf{P}|_{D|^{\mathbf{x}}} \| \, . \tag{12}$$

where the max is restricted to $\mathbf{x} \in X^k$. Obviously, $\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \rrbracket = \llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}' \big| \mathcal{X} \rrbracket$.

Lemma 6. *Let* $X = X' \cup X'' \subseteq \mathcal{X}$ *and* $k = k' + k''$*. Furthermore, let* $\mathsf{P}, \mathsf{P}', \mathsf{P}''$ *and* Q *be database properties. Then*

$$\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}'' \big| X \rrbracket \leq \llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{P}'' \backslash \mathsf{Q} \big| X \rrbracket + \llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{Q} \cap \mathsf{P}'' \big| X \rrbracket \, ,$$

where furthermore

$$\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{Q} \cap \mathsf{P}'' \big| X \rrbracket \leq \llbracket \mathsf{P} \overset{k'}{\rightarrow} \neg \mathsf{Q} \big| X \rrbracket + \llbracket \mathsf{P} \overset{k'}{\rightarrow} \mathsf{Q} \cap \mathsf{P}' \big| X \rrbracket + \llbracket \mathsf{Q} \backslash \mathsf{P}' \overset{k''}{\rightarrow} \mathsf{Q} \cap \mathsf{P}'' \big| X \rrbracket$$

as well as

$$\llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{Q} \cap \mathsf{P}'' \big| X \rrbracket \leq \llbracket \mathsf{P} \overset{k}{\rightarrow} \neg \mathsf{Q} \big| X' \rrbracket + \llbracket \mathsf{P} \overset{k}{\rightarrow} \mathsf{Q} \cap \mathsf{P}' \big| X' \rrbracket + \llbracket \mathsf{Q} \backslash \mathsf{P}' \overset{k}{\rightarrow} \mathsf{Q} \cap \mathsf{P}'' \big| X'' \rrbracket .$$

By recursive application of Lemma 6, we obtain the following.

Corollary 3 (Parallel Conditioning). *Let* $X = X_1 \cup \ldots \cup X_h \subseteq \mathcal{X}$ *and* $k = k_1 + \cdots + k_h$*, and let* $\mathsf{P}_0, \mathsf{P}_1, \ldots, \mathsf{P}_h$ *and* $\neg \mathsf{P}_0 \subseteq \mathsf{Q}$ *be database properties. Then*

$$\llbracket \neg \mathsf{P}_0 \overset{k}{\rightarrow} \mathsf{P}_h \big| X \rrbracket \leq \sum_{i=1}^{h} \llbracket \neg \mathsf{P}_0 \overset{\bar{k}_i}{\rightarrow} \neg \mathsf{Q} \big| X \rrbracket + \sum_{i=1}^{h} \llbracket \mathsf{Q} \backslash \mathsf{P}_{i-1} \overset{k_i}{\rightarrow} \mathsf{Q} \cap \mathsf{P}_i \big| X \rrbracket \qquad \textit{and}$$

$$\llbracket \neg \mathsf{P}_0 \overset{k}{\rightarrow} \mathsf{P}_h \big| X \rrbracket \leq \sum_{i=1}^{h} \llbracket \neg \mathsf{P}_0 \overset{k}{\rightarrow} \neg \mathsf{Q} \big| \bar{X}_i \rrbracket + \sum_{i=1}^{h} \llbracket \mathsf{Q} \backslash \mathsf{P}_{i-1} \overset{k}{\rightarrow} \mathsf{Q} \cap \mathsf{P}_i \big| X_i \rrbracket \, ,$$

where $\bar{k}_i = k_1 + \cdots + k_i$ *and* $\bar{X}_i = X_1 \cup \ldots \cup X_i$*.*

The quantum transition capacity *with restricted input*, defined in (12), is just the original definition of the quantum transition capacity (Definition 2) but with the considered set $\mathcal{X}$ replaced by X. As a consequence, properties for $[\![\mathsf{P} \to \mathsf{P}']\!]$ carry over to $[\![\mathsf{P} \to \mathsf{P}' | X]\!]$. For instance, it is still symmetric, and Lemma 5 carries over to

$$[\![\mathsf{P} \cap \mathsf{Q} \xrightarrow{k} \mathsf{P}' | X]\!] \leq \min\{ [\![\mathsf{P} \xrightarrow{k} \mathsf{P}' | X]\!], [\![\mathsf{Q} \xrightarrow{k} \mathsf{P}' | X]\!] \}$$

etc. For completeness, we spell out here the definition of non-uniform recognizability as well as Theorem 3 for such input-restricted database transitions $\mathsf{P} \to \mathsf{P}' \,|\, X$ (the other types of recognizability can be generalized similarly).

Definition 8. *A database transition* $\mathsf{P} \to \mathsf{P}'$ *with input restricted in* $X \subseteq \mathcal{X}$ *is said to be* k-non-uniformly weakly recognizable *by* ℓ*-local properties if for every* $\mathbf{x} = (x_1, \ldots, x_k) \in X^k$ *with disjoint entries, and for every* $D \in \mathfrak{D}$*, there exist a family of* ℓ*-local properties* $\{\mathsf{L}_i^{\mathbf{x},D}\}_i$ *with supports in* $\{x_1, \ldots, x_k\}$ *so that*

$$D_\circ \in \mathsf{P}|_{D|^{\mathbf{x}}} \wedge D' \in \mathsf{P}'|_{D|^{\mathbf{x}}} \implies \exists\, i \colon D' \in \mathsf{L}_i^{\mathbf{x},D} \wedge \big(\exists\, x \in \mathsf{Supp}(\mathsf{L}_i^{\mathbf{x},D}) : D_\circ(x) \neq D'(x)\big).$$

Theorem 7. *Let* $\mathsf{P} \to \mathsf{P}'$ *with input restricted in* X *be* k*-non-uniformly weakly recognizable by* 1*-local properties* $\mathsf{L}_i^{\mathbf{x},D}$*, where the support of* $\mathsf{L}_i^{\mathbf{x},D}$ *is* $\{x_i\}$ *or empty. Then*

$$[\![\mathsf{P} \xrightarrow{k} \mathsf{P}' | X]\!] \leq \max_{\mathbf{x},D} e \sum_i \sqrt{10 P\big[U \in \mathsf{L}_i^{\mathbf{x},D}\big]}\,,$$

where the max *now is over all* $\mathbf{x} = (x_1, \ldots, x_k) \in X^k$.

6 Post-Quantum Proof of Sequential Works

In this section, we prove post-quantum security of the proof of sequential work (PoSW) construction by Cohen and Pietrzak [10] (referred to as Simple PoSW) using our framework developed in the last section. As a matter of fact, we directly analyze the non-interactive variant of their construction after applying the Fiat-Shamir transformation [12]. As we shall see, the proof is by means of purely classical reasoning, recycling observations that are relevant for arguing classical security and combining them with results provided by our framework.

6.1 Simple Proof of Sequential Works

For readers not familiar with PoSW, we review the definition in the full version [9]. Typically, underlying the construction of a PoSW is a directed acyclic graph (DAG) G with certain "depth-robust" properties, and a graph labeling that the prover $\mathcal{P}$ is required to compute using a hash function H. We proceed to describe the DAG used in Simple PoSW and the graph labeling.

Simple PoSW DAG and Graph Labeling. Let $n \in \mathbb{N}$ and $N = 2^{n+1} - 1$. Consider the (directed) complete binary tree $B_n = (V_n, E'_n)$ of depth n, where $V_n := \{0,1\}^{\leq n}$ and E'_n consists of the edges directed towards the root (black edges in Fig. 2). The Simple PoSW DAG, denoted by G_n^{PoSW}, is obtained by adding some additional edges to B_n (red edges in Fig. 2). Before giving the formal definition of G_n^{PoSW} (Definition 10), we recall some basic terminology and notation in the context of the complete binary tree B_n, which we will then also use in the context of G_n^{PoSW}.

Definition 9. *We write* $\mathsf{rt} := \epsilon$ *for the* root *and* $\mathsf{leaves}(V_n) := \{0,1\}^n$ *for the* leaves *in* V_n*. For* $T \subseteq V_n$*, we set* $\mathsf{leaves}(T) := T \cap \{0,1\}^n$*. For* $v \notin \mathsf{leaves}(V_n)$*, we set* $\mathsf{left}(v) := v\|0$ *and* $\mathsf{right}(v) := v\|1$*. For* $b \in \{0,1\}$ *and* $v \in \{0,1\}^{<n}$*, let* $\mathsf{par}(v\|b) := v$ *and* $\mathsf{sib}(v\|b) := v\|\neg b$ *(see Fig. 2, right).*

Finally, for a leaf $v \in \mathsf{leaves}(V_n)$*, we define the* ancestors *of* v *as* $\mathsf{anc}(v) = \{\mathsf{par}^i(v) \mid 0 \leq i \leq n\}$ *and the* authentication path *of* v *(as in the Merkle tree) as* $\mathsf{ap}(v) = (\mathsf{anc}(v) \setminus \{\mathsf{rt}\}) \cup \{\mathsf{sib}(u) \mid \mathsf{rt} \neq u \in \mathsf{anc}(v)\}$.

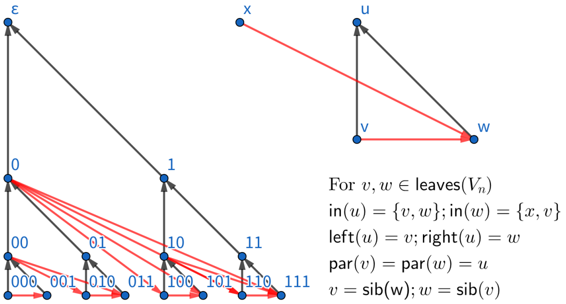
ε
x
u
v
w
0
1
00
01
10
11
000
001
010
011
100
101
110
111
For $v, w \in \mathsf{leaves}(V_n)$
$\mathsf{in}(u) = \{v, w\}; \mathsf{in}(w) = \{x, v\}$
$\mathsf{left}(u) = v; \mathsf{right}(u) = w$
$\mathsf{par}(v) = \mathsf{par}(w) = u$
$v = \mathsf{sib}(w); w = \mathsf{sib}(v)$

Fig. 2. Illustration of the Simple PoSW DAG G_n^{PoSW} for $n = 3$.

Definition 10. *Define the Simple PoSW DAG* $G_n^{\mathsf{PoSW}} := (V_n, E'_n \cup E''_n)$ *with*

$$E'_n := \{(\mathsf{left}(v), v), (\mathsf{right}(v), v) \mid v \in V_n \setminus \mathsf{leaves}(V_n)\} \quad \textit{and}$$
$$E''_n := \{(\mathsf{sib}(u), v) \mid v \in V_n, u \in \mathsf{anc}(v) \;\; \textit{s.t.} \;\; u = \mathsf{right}(\mathsf{par}(u))\}\,.$$

For $v \in V_n$, we write $\mathsf{in}(v) := \{u \in V_n \mid (u,v) \in E'_n \cup E''_n\}$ to denote the inward neighborhood of v. We consider a fixed ordering of the vertices (e.g. lexicographic), so that for any set $\{v_1, \ldots, v_d\} \in V_n$ of vertices, the corresponding ordered list $(v_1, \ldots, v_d)$ is well defined.

We proceed to define the graph labeling for G_n^{PoSW} with respect to a hash function $H : \{0,1\}^{\leq B} \to \{0,1\}^w$, were w is a security parameter, and B is arbitrary large (and sufficiently large for everything below to be well defined).

Definition 11 (Graph Labeling). *A function* $\ell : V_n \to \{0,1\}^w$*,* $v \mapsto \ell_v$ *is a* labeling *of* G_n^{PoSW} *with respect to* H *if*

$$\ell_v = H(v, \ell_{\mathsf{in}(v)}) \tag{13}$$

for all $v \in V_n$*, were* $\ell_{\mathsf{in}(v)}$ *is shorthand for* $(\ell_{v_1}, \dots, \ell_{v_d})$ *with* $\{v_1, \dots, v_d\} = \mathsf{in}(v)$*. Similarly, for a subtree*[10] T *of* G_n^{PoSW}*, a function* $\ell : T \to \{0,1\}^w$*,* $v \mapsto \ell_v$ *is a called a* labeling *of* T *with respect to* H *if* $\ell_v = H(v, \ell_{\mathsf{in}(v)})$ *for all* $v \in V_n$ *for which* $\mathsf{in}(v) \subseteq T$*.*

By its structure, G_n^{PoSW} admits a unique labeling, which can be computed by making $N = 2^{n+1} - 1$ sequential queries to H, starting with the leftmost leaf. We speak of a *consistent* labeling (of G_n^{PoSW} or T) when we want to emphasize the distinction from an arbitrary function ℓ. The definition also applies when replacing the function H by a database $D : \{0,1\}^{\leq B} \to \{0,1\}^w \cup \{\bot\}$, where the requirement (13) then in particular means that $H(v, \ell_{\mathsf{in}(v)}) \neq \bot$.

We also make the following important remark.

Remark 9. Let T be a subtree of G_n^{PoSW} with a consistent labeling ℓ. Then, any path $P = (v_0, \dots, v_r)$ of length $|P| = r$ in T induces an r-chain $(x_0, \dots, x_r)$, where $x_i = (v_i, \ell_{v'_1}, \dots, \ell_{v'_d})$ with $\{v'_1, \dots, v'_d\} = \mathsf{in}(v_i)$, and where the relation $\lhd$ is defined as follows. $y \lhd x$ if and only if x is of form $(v, \ell_1, \ell_2, \dots, \ell_d)$ with $v \in V_n, \ell_j \in \{0,1\}^w$, $|d| = |\mathsf{in}(v)| \leq n$, and $y = \ell_j$ for some j.

Simple PoSW Construction. In the Simple PoSW scheme, the prover $\mathcal{P}$ must provide the root label ℓ_{rt} of the consistent labeling of G_n^{PoSW} with respect to H_χ defined by $H_\chi(\cdot) := H(\chi, \cdot)$ for a random $\chi \in \{0,1\}^w$, sampled by the verifier $\mathcal{V}$, and open the labels of the authentication paths of a few random leaves.

Specifically, given parameters w, t and $N = 2^{n+1} - 1$, and a random oracel $H : \{0,1\}^{\leq B} \to \{0,1\}^w$, the Simple PoSW protocol is defined as follows.

- $(\phi, \phi_{\mathcal{P}}) := \mathsf{PoSW}^H(\chi, N)$: $\mathcal{P}$ computes the unique consistent labeling ℓ of G_n^{PoSW} with respect to hash function H_χ defined by $H_\chi(\cdot) := H(\chi, \cdot)$, and stores it in $\phi_{\mathcal{P}}$. $\mathcal{P}$ sets $\phi = \ell_{\mathsf{rt}}$ as the root label.
- The opening challenge: $\gamma := H_\chi^{\mathsf{ChQ}}(\phi) := \big(H_\chi(\phi, 1), \dots, H_\chi(\phi, d)\big) \in \{0,1\}^{dw}$ for sufficiently large d, parsed as t leaves $\{v_1, \dots, v_t\} \subseteq \mathsf{leaves}(V_n)$.
- $\tau := \mathsf{open}^H(\chi, N, \phi_{\mathcal{P}}, \gamma)$: For challenge $\gamma = \{v_1, \dots, v_t\}$, the opening τ consists of the labels of the vertices in the authentication path $\mathsf{ap}(v_i)$ of v_i for $i \in [t]$, i.e., $\tau = \{\ell_{\mathsf{ap}(v_i)}\}_{i \in [t]}$.
- $\mathsf{verify}^H(\chi, N, \phi, \gamma, \tau)$: $\mathcal{V}$ verifies if the ancestors of every v_i are consistently labeled by τ. Specifically, for each $i \in [t]$, $\mathcal{V}$ checks if $\ell_u = H_\chi(u, \ell_{\mathsf{in}(u)})$ for all $u \in \mathsf{anc}(v_i)$. $\mathcal{V}$ outputs accept iff all the consistency checks pass.

Note that since we consider the non-interactive version of Simple PoSW after applying the Fair-Shamir transformation, the random oracle H is used to compute both the labels (as $H_\chi(v, \ell_{\mathsf{in}(v)})$) and the challenge (as $H_\chi^{\mathsf{ChQ}}(\phi)$). We silently assume that the respective inputs are specially formatted so as to distinguish a *label query* from a *challenge query*. E.g., a label query comes with a

[10] By a *subtree* of G_n^{PoSW} we mean a sub*graph* of G_n^{PoSW} that is a sub*tree* of the complete binary tree B_n when restricted to edges in E'_n. We are also a bit sloppy with not distinguishing between the graph T and the vertices of T.

prefix 0 and a challenge query with prefix 1. We then denote the set of inputs for label and challenge queries by LbQ and $\mathsf{ChQ} \subseteq \{0,1\}^{\leq B}$, respectively. Also, for simplicity, we will treat $H_\chi^{\mathsf{ChQ}}(\phi)$ as *one* oracle query, i.e., "charge" only one query for a challenge query; however, we keep the superscript ChQ to remind that the query response is (understood as) a set of leaves.

Classical Security Analysis of Simple PoSW. We first review the classical security analysis of [10]. For simplicity, here we consider the original (interactive) variant (i.e., $\mathcal{P}$ first sends ϕ, receives random γ from $\mathcal{V}$, and then sends τ to $\mathcal{V}$). Also, to start with, we assume that $\mathcal{P}$ does not make further oracle queries after sending ϕ. We review the argument of [10] for bounding the probability that a k-parallel q-query classical oracle algorithm $\mathcal{A}$ with $q < N$ makes $\mathcal{V}$ accept, using the terminology we introduced in Section 2.

Let $D : \{0,1\}^{\leq B} \to \{0,1\}^w \cup \{\perp\}$ be the database at the point that $\mathcal{A}$ sends ϕ to $\mathcal{V}$. Following the argument in Section 2, we can bound the success probability of $\mathcal{A}$ by the probability that a random challenge $\gamma = \{v_i\}_{i\in[t]}$ can be opened based on the information in the database D.

First, since the probability is small for the database D to contain a collision or a $(q+1)$-chain with respect to the relation defined in Remark 9, we can assume that D contains no collisions nor $(q+1)$-chains.

Next, given the database D and the "commitment" ϕ, claimed to be the root label ℓ_{rt}, we need to analyze the set of leaves v that $\mathcal{A}$ can open. One of the key observations in [10] is that, for a database D with no collisions, there exists a maximal subtree T of G_n^{PoSW} that contains rt and admits a consistent labeling ℓ with $\ell_{\mathsf{rt}} = \phi$. As observed in [10], this subtree T then contains all leaves that one can open given D. Thus, $\mathcal{A}$ can correctly answer a challenge $\gamma = \{v_1, \dots, v_t\}$ if $\gamma \subseteq \mathsf{leaves}(T)$, while otherwise it is unlikely that he succeeds.

The subtree T, together with the labeling ℓ of T, can be extracted using an algorithm $\mathsf{Extract}_n^D(\phi)$, described in the full version [9]. Roughly speaking, starting with $T := \{\mathsf{rt}\}$, consider $v := \mathsf{rt}$ and $\ell_{\mathsf{rt}} := \phi$, and add $\mathsf{left}(v)$ and $\mathsf{right}(v)$ to T if (and only if) there exist $\ell_{\mathsf{left}(v)}$ and $\ell_{\mathsf{right}(v)}$ such that $\ell_v = D\big(v, \ell_{\mathsf{left}(v)}, \ell_{\mathsf{right}(v)}\big)$, and repeat inductively with the newly added elements in T. In the end, for the leaves $v \in T$ check if $\ell_v = D(v, \ell_{\mathsf{in}(v)})$ and remove v from T if this is not the case; we note here that $v \in \mathsf{leaves}(T) \Rightarrow \mathsf{in}(v) \subseteq T$.

Lemma 7. *Let $D : \{0,1\}^{\leq B} \to \{0,1\}^w \cup \{\perp\}$ be a database with no collisions (beyond $\perp$). Then, for any $\phi \in \{0,1\}^w$, the subtree T and the labeling ℓ produced by $\mathsf{Extract}_n^D(\phi)$ are such that ℓ is a consistent labeling of T with respect to D, having root label $\ell_{\mathsf{rt}} = \phi$. Furthermore, for any leave v of V_n, if $v \in T$ then $\ell_u = D(u, \ell_{\mathsf{in}(u)})$ for all $u \in \mathsf{anc}(v_i)$, and if $v \notin T$ then there exists no labeling ℓ' with $\ell'_{\mathsf{rt}} = \phi$ and $\ell'_u = D(u, \ell'_{\mathsf{in}(u)})$ for all $u \in \mathsf{anc}(v_i)$.*

Another key argument in [10] uses a certain "depth-robust" property of G_n^{PoSW} to show that for any subtree $T \subseteq V_n$ with $\mathsf{rt} \in T$, there exists a path P in T with length $|P| \geq 2\cdot|\mathsf{leaves}(T)|-2$. Furthermore, Remark 9 applies here as well: a labeling of P with respect to D induces a $|P|$-chain in D. Combining these with the assumption that D contains no $q+1$-chain, we have $|\mathsf{leaves}(T)| \leq (q+2)/2$. Thus, the probability that $\mathcal{A}$ can open labels for a random challenge $\gamma = \{v_i\}_{i\in[t]}$ is at most

$$\left(\frac{|\mathsf{leaves}(T)|}{2^n}\right)^t \leq \left(\frac{q+2}{2^{n+1}}\right)^t .$$

Lemma 8. *Let $D : \{0,1\}^{\leq B} \to \{0,1\}^w \cup \{\bot\}$ be a database with no $(q+1)$-chain. Let T be a subtree of G_n^{PoSW} admitting a consistent labeling with respect to D. Then, $|\mathsf{leaves}(T)| \leq (q+2)/2$.*

Finally, we briefly discuss here how to deal with $\mathcal{A}$ making additional queries after sending ϕ. Recall that T contains all leaves v that admit consistently labeled ancestors. Thus, for the additional queries to be helpful, they must enlarge the extracted subtree T. More precisely, let D' be the database after the additional queries and let T' and ℓ' be extracted by $\mathsf{Extract}_n^{D'}(\phi)$. It must be that $T \subsetneq T'$ and $\ell'|_T = \ell$, and there must exist x with $D(x) = \bot$ while $D'(x) = \ell_v$ for some $v \in T$. This happens with probability at most $O(qk/2^w)$ for each query since ℓ has support size at most $O(qk)$. We capture the above crucial observation by means of the following formal statement, which, in this form, will then be recycled in the security proof against quantum attacks.

Lemma 9. *Let $D : \{0,1\}^{\leq B} \to \{0,1\}^w \cup \{\bot\}$ be a database with no collisions (beyond $\bot$). Let $\phi \in \{0,1\}^w$ and $(T,\ell) = \mathsf{Extract}_n^D(\phi)$. Furthermore, let $D' = D[\mathbf{x} \mapsto \mathbf{u}]$ and $(T',\ell') = \mathsf{Extract}_n^{D'}(\phi)$, and let v be a leave of V_n. If $v \in T' \setminus T$ then there exist $j \in \{1,\dots,k\}$ and $z \in \mathsf{anc}(v)$ so that $D(x_j) \neq D'(x_j) = \ell'_z$.*

Proof. Given that $v \in T'$, the labeling ℓ' labels the ancestors of v consistently with respect to D', i.e., $\ell'_z = D'(z, \ell'_{\mathsf{in}(z)})$ for all $z \in \mathsf{anc}(v)$. On the other hand, as $v \notin T$, ℓ' does *not* label the ancestors of v consistently with respect to D, i.e., there must exist $z \in \mathsf{anc}(v)$ such that $D(z, \ell'_{\mathsf{in}(z)}) \neq \ell'_z = D'(z, \ell'_{\mathsf{in}(z)})$. Since D and D' differ only within $\mathbf{x}$, there exists $j \in \{1,\dots,k\}$ with $x_j = (z, \ell'_{\mathsf{in}(z)})$. □

6.2 Post-Quantum Security of Simple PoSW

In this section, we prove post-quantum security of the (non-interactive) Simple PoSW protocol. As we shall see, relying on the framework we developed in Section 5, the proof uses *purely classical reasoning* only, and somewhat resembles the arguments in the classical analysis.

Theorem 8 (Post-Quantum Simple PoSW Security). *Consider the Simple PoSW protocol with parameters w, t and $N = 2^{n+1} - 1$ with $w \geq tn$. Let $\tilde{\mathcal{P}}$ be a k-parallel q-query quantum oracle algorithm acting as a prover. The probability p that $\tilde{\mathcal{P}}$ can make the verifier $\mathcal{V}$ accept is at most*

$$p = O\left(k^2 q^2 \left(\frac{q+2}{2^{n+1}}\right)^t + \frac{k^3 q^3 n}{2^w} + \frac{tn}{2^w}\right).$$

The first step towards the proof is to invoke Corollary 1 (using the notation from Theorem 1), which, in the case here, bounds the success probability p of a dishonest prover $\tilde{\mathcal{P}}$ by

$$\sqrt{p} \leq [\![\bot \overset{k,q}{\Longrightarrow} \mathsf{P}^R]\!] + \sqrt{\frac{t \cdot (n+1) + 1}{2^w}} \,,$$

where R is the relation that checks correctness of $\tilde{\mathcal{P}}$'s output according to the scheme. In the following, we write $\mathsf{Suc} := \mathsf{P}^R$ and $\mathsf{Fail} = \neg\mathsf{Suc}$. Also, recall the database properties CL, $\mathsf{SZ}_{\leq s}$ and CHN^s defined previously, where the latter is with respect to the hash chain relation $\lhd$ considered in Remark 9. By the properties of (the subtree extracted with) $\mathsf{Extract}_n^D(\cdot)$, we have

$$\mathsf{Suc} \setminus \mathsf{CL} = \big\{ D \in \neg\mathsf{CL} \,\big|\, \exists\, \ell_{\mathsf{rt}} \in \{0,1\}^w \text{ s.t. } D^{\mathsf{ChQ}}(\ell_{\mathsf{rt}}) \subseteq \mathsf{Extract}_n^D(\ell_{\mathsf{rt}}) \big\} \,. \tag{14}$$

To bound $[\![\bot \overset{k,q}{\Longrightarrow} \mathsf{P}^R]\!] = [\![\bot \overset{k,q}{\Longrightarrow} \mathsf{Suc}]\!]$, we consider database properties $\mathsf{P}_0, \ldots, \mathsf{P}_q$ with $\mathsf{P}_0 = \bot$ and $\mathsf{P}_s = \mathsf{Suc} \cup \mathsf{CL} \cup \mathsf{CHN}^{s+1}$ for $1 \leq s \leq q$. Using Lemma 4, Remark 3 and Corollary 2,

$$[\![\bot \overset{k,q}{\Longrightarrow} \mathsf{Suc}]\!] \leq \sum_{1 \leq s \leq q} [\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s]\!] \,.$$

Thus, the proof of Theorem 8 follows immediately from the following bound.

Proposition 1. *For integers $0 \leq s \leq q$, and for the database properties $\mathsf{P}_0, \ldots, \mathsf{P}_q$ as defined above*

$$[\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s]\!] \leq 4ek\sqrt{10\frac{q+1}{2^w}} + 3ek\sqrt{\frac{10kqn}{2^w}} + ek\sqrt{10\left(\frac{q+2}{2^{n+1}}\right)^t} \,.$$

Proof. By applying Corollary 3 with $h := 2$, $X_1 := \mathsf{LbQ}$ and $X_2 := \mathsf{ChQ}$, and with $\mathsf{P}_0, \mathsf{P}_1, \mathsf{P}_2$ and Q suitably chosen (we have to refer to the full version [9] for the details here), we obtain

$$\begin{aligned} [\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s]\!] &\leq 2[\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{CL} \cup \mathsf{CHN}^{s+1}]\!] \\ &\quad + [\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{Suc} \backslash \mathsf{CL} \big| \mathsf{LbQ}]\!] + [\![\neg\mathsf{P}_s \xrightarrow{k} \mathsf{Suc} \backslash \mathsf{CL} \big| \mathsf{ChQ}]\!] \,. \end{aligned}$$

By Lemma 5 (and Corollary 2), and recalling that $\mathsf{P}_{s-1} = \mathsf{Suc} \cup \mathsf{CL} \cup \mathsf{CHN}^s$, the first capacity in the term can be controlled as

$$\begin{aligned} &[\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{CL} \cup \mathsf{CHN}^{s+1}]\!] \\ &\qquad \leq [\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{CL}]\!] + [\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{CHN}^{s+1}]\!] \\ &\qquad \leq [\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{CL} \xrightarrow{k} \mathsf{CL}]\!] + [\![\mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{CHN}^s \xrightarrow{k} \mathsf{CHN}^{s+1}]\!] \\ &\qquad \leq 2ek\sqrt{10\frac{q+1}{2^w}} + ek\sqrt{\frac{10kqn}{2^w}} \end{aligned}$$

using earlier derived bounds. It remains to bound the remaining two capacities appropriately, which we do below. □

Lemma 10. *For any* $0 < q \in \mathbb{Z}$*:* $[\![\neg \mathsf{P}_q \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL}\big|\mathsf{ChQ}]\!] \le ek \cdot \sqrt{10\left(\frac{q+2}{2^{n+1}}\right)^t}$.

Proof. For convenience, we will denote $D[\mathbf{x} \mapsto \mathbf{y}]$ by $D_{\mathbf{x},\mathbf{y}}$. In order to bound the above capacity, we define 1-local properties $\mathsf{L}_j^{\mathbf{x},D}$ and show that $\mathsf{L}_j^{\mathbf{x},D}$ (weakly) recognize the considered transition (with input restricted to ChQ).

For any D and $\mathbf{x} = (\ell_{\mathsf{rt}}^1, \ldots, \ell_{\mathsf{rt}}^k) \in \mathsf{ChQ}^k$, we set

$$\mathsf{L}_j^{\mathbf{x},D} := \Big\{ D_\circ \in D|^{\mathbf{x}} \;\Big|\; D_\circ^{\mathsf{ChQ}}(x_j) \subseteq \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\perp}}(\ell_{\mathsf{rt}}^j)\Big)\Big\}$$

Suppose $D_{\mathbf{x},\mathbf{r}} \in \neg\mathsf{P}_q = \mathsf{Fail} \setminus \mathsf{CL} \setminus \mathsf{CHN}^{q+1}$ but $D_{\mathbf{x},\mathbf{u}} \in \mathsf{Suc} \setminus \mathsf{CL}$. Thus, by (14), there exists $\ell_{\mathsf{rt}} \in \{0,1\}^w$ with

$$D_{\mathbf{x},\mathbf{u}}^{\mathsf{ChQ}}(\ell_{\mathsf{rt}}) \subseteq \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{u}}}(\ell_{\mathsf{rt}})\Big)\,, \tag{15}$$

while

$$D_{\mathbf{x},\mathbf{r}}^{\mathsf{ChQ}}(\ell_{\mathsf{rt}}) \not\subseteq \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{r}}}(\ell_{\mathsf{rt}})\Big). \tag{16}$$

Since the output of the extraction procedure $\mathsf{Extract}_n^D(\cdot)$ only depends on those function values of D that correspond to *label* queries ($\mathbf{x}$ here consists of challenge queries), we have

$$\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{r}}}(\ell_{\mathsf{rt}}) = \mathsf{Extract}_n^{D_{\mathbf{x},\perp}}(\ell_{\mathsf{rt}}) = \mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{u}}}(\ell_{\mathsf{rt}}).$$

If ℓ_{rt} is different from all ℓ_{rt}^j, then equations (15) and (16) contradict. So there is some j such that $\ell_{\mathsf{rt}}^j = \ell_{\mathsf{rt}}$. Equations (15) and (16) thus become

$$u_j \subseteq \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\perp}}(\ell_{\mathsf{rt}})\Big) \qquad \text{and} \qquad r_j \not\subseteq \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\perp}}(\ell_{\mathsf{rt}})\Big),$$

understanding that u_j and r_j represent lists/sets of t (challenge) leaves. Hence $r_j \neq u_j$. This concludes that $\mathsf{L}_j^{\mathbf{x},D}$ indeed weakly recognizes the considered database transition.

We note that, for each $\mathbf{x} \in \mathsf{ChQ}^k$ and $D \in \mathsf{Fail}\setminus\mathsf{CL}\setminus\mathsf{CHN}^{q+1}$, since the longest hash chain in D is of length no more than q and $T := \mathsf{Extract}_n^{D_{\mathbf{x},\perp}}(\ell_{\mathsf{rt}}^j)$ admits a consistent labeling (Lemma 7), it follows from Lemma 8 that the number of leaves in T is bounded by $(q+2)/2$. Therefore,

$$P\big[U \in \mathsf{L}_j^{\mathbf{x},D}\big] \le \left(\frac{\mathsf{leaves}\big(\mathsf{Extract}_n^{D_{\mathbf{x},\perp}}(\ell_{\mathsf{rt}}^j)\big)}{2^n}\right)^t \le \left(\frac{q+2}{2^{n+1}}\right)^t\,,$$

and so the claimed bound follows by applying Theorem 7. □

Lemma 11. *For any* $0 < q \in \mathbb{Z}$*:* $[\![\mathsf{SZ}_{\le k(q-1)}\backslash\mathsf{P}_{q-1} \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL}\big|\mathsf{LbQ}]\!] \le ek\sqrt{\frac{10nkq}{2^w}}$.

Proof. Define the notion of labeling support $\mathrm{LSupp}(D)$ of a database $D \in \mathfrak{D}$ as follows.

$$\mathrm{LSupp}(D) := \left\{ \lambda \in \{0,1\}^w \;\middle|\; \begin{array}{l} \exists\, 0 \le i \le d \le n, v \in V_n, \ell_1, \ldots, \ell_d \in \{0,1\}^w \\ \text{s.t. } D(v, \ell_1, \ldots, \ell_{i-1}, \lambda, \ell_{i+1}, \ldots \ell_d) \neq \perp \end{array} \right\}$$

$$\cup \left\{ \ell_{\mathsf{rt}} \in \{0,1\}^w \;\middle|\; D^{\mathsf{ChQ}}(\ell_{\mathsf{rt}}) \neq \perp \right\}.$$

We note that since LSupp is defined only in terms of where D is defined, but does not depend on the actual function values (beyond being non-$\perp$), $\mathrm{LSupp}(D) \subseteq \mathrm{LSupp}(D_{\mathbf{x},\mathbf{0}})$ for any $\mathbf{x} \in \mathcal{X}^k$, where $\mathbf{0} \in \{0,1\}^k$ is the all-0 string.

In order to bound above capacity, we define 1-local properties and show that they (weakly) recognize the considered transition (with input restricted to LbQ). For any D and $\mathbf{x} \in \mathsf{LbQ}^k$, consider the local properties

$$\mathsf{L}_j^{\mathbf{x},D} := \left\{ D_\circ \in D|^{\mathbf{x}} \;\middle|\; D_\circ(x_j) \in \mathrm{LSupp}(D_{\mathbf{x},\mathbf{0}}) \right\} .$$

Let $D_{\mathbf{x},\mathbf{r}} \in \neg\mathsf{P}_{q-1} = \mathsf{Fail} \setminus \mathsf{CL} \setminus \mathsf{CHN}^q$ yet $D_{\mathbf{x},\mathbf{u}} \in \mathsf{Suc} \setminus \mathsf{CL}$. By (14), there exists ℓ_{rt} so that $D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{u}}(\ell_{\mathsf{rt}}) \subseteq \mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{u}}}(\ell_{\mathsf{rt}})$, while, on the other hand, there exists some $v \in D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{r}}(\ell_{\mathsf{rt}}) \setminus \mathsf{leaves}\big(\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{r}}}(\ell_{\mathsf{rt}})\big)$. Given that here $\mathbf{x} \in \mathsf{LbQ}^k$, we have $D_{\mathbf{x},\mathbf{r}}(\ell_{\mathsf{rt}}) = D_{\mathbf{x},\mathbf{u}}(\ell_{\mathsf{rt}})$, and thus, by (15), we have

$$v \in \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{u}}}(\ell_{\mathsf{rt}})\Big) \setminus \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{r}}}(\ell_{\mathsf{rt}})\Big).$$

Writing ℓ' for the labeling extracted by $\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{u}}}(\ell_{\mathsf{rt}})$, it then follows from Lemma 9 that there exist $j \in \{1,\dots,k\}$ and $z \in \mathsf{anc}(v)$ such that $u_j = D_{\mathbf{x},\mathbf{u}}(x_j) = \ell'_z \neq D_{\mathbf{x},\mathbf{r}}(x_j) = r_j$. Furthermore, since $D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{u}}(\ell'_z) = D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{u}}(\ell_{\mathsf{rt}}) \neq \perp$ in case $z = \mathsf{rt}$, and ℓ'_z is part of the input that is mapped to $\ell'_{\mathsf{par}(z)}$ under $D_{\mathbf{x},\mathbf{u}}$ in all other cases, we also have $u_j = \ell'_z \in \mathrm{LSupp}(D_{\mathbf{x},\mathbf{u}}) \subseteq \mathrm{LSupp}(D_{\mathbf{x},\mathbf{0}})$. Therefore, the local properties $\mathsf{L}_j^{\mathbf{x},D}$ do indeed weakly recognize the considered transition for input restricted to LbQ.

For $D \in \mathsf{SZ}_{\leq k(q-1)} \setminus \mathsf{P}_{q-1}$, since there are only $k(q-1)$ entries in D, we have

$$P[U \in \mathsf{L}_j^{\mathbf{x},D}] \leq \frac{|\mathrm{LSupp}(D_{\mathbf{x},\mathbf{0}})|}{2^w} \leq \frac{nkq}{2^w}. ,$$

and thus the claimed bound follows from applying Theorem 7. □

Acknowledgements

We thank Jeremiah Blocki, Seunghoon Lee, and Samson Zhou for the open discussion regarding their work [4], which achieves comparable results for the hash-chain problem and the Simple PoSW scheme.

References

1. Ambainis, A.: Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. Theory of Computing 1(1), 37–46 (2005)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: First ACM Conference on Computer and Communications Security. pp. 62–73. ACM (1993)
3. Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.: Strengths and weaknesses of quantum computing. SIAM journal on Computing 26(5), 1510–1523 (1997)

4. Blocki, J., Lee, S., Zhou, S.: On the security of proofs of sequential work in a post-quantum world. arXiv/cs.CR, Report 2006.10972 (2020), https://arxiv.org/abs/2006.10972
5. Boneh, D., Dagdelen, O., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: D.H., L., X., W. (eds.) Advances in Cryptology ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer (2011)
6. Brassard, G., Hoyer, P., Tapp, A.: Quantum algorithm for the collision problem. arXiv/quant-ph, Report 9705002 (1997), https://arxiv.org/abs/quant-ph/9705002
7. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. Journal of the ACM (JACM) 51(4), 557–594 (2004)
8. Chiesa, A., Manohar, P., Spooner, N.: Succinct arguments in the quantum random oracle model. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography - TCC 2019. Lecture Notes in Computer Science, vol. 11892. Springer (2019)
9. Chung, K.M., Fehr, S., Huang, Y.H., Liao, T.N.: On the compressed-oracle technique, and post-quantum security of proofs of sequential work. Cryptology ePrint Archive, Report 2020/1305 (2020), https://eprint.iacr.org/2020/1305
10. Cohen, B., Pietrzak, K.: Simple proofs of sequential work. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 451–467. Springer (2018)
11. Czajkowski, J., Majenz, C., Schaffner, C., Zur, S.: Quantum lazy sampling and game-playing proofs for quantum indifferentiability. arXiv/quant-ph, Report 1904.11477 (2019), https://arxiv.org/abs/1904.11477
12. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Conference on the Theory and Application of Cryptographic Techniques. pp. 186–194. Springer (1986)
13. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)
14. Hamoudi, Y., Magniez, F.: Quantum time-space tradeoffs by recording queries. arXiv/quant-ph, Report 2002.08944 (2020), https://arxiv.org/abs/2002.08944
15. Hosoyamada, A., Iwata, T.: 4-round luby-rackoff construction is a qprp. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019. Lecture Notes in Computer Science, vol. 11921, pp. 145–174. Springer (2019)
16. Jeffery, S., Magniez, F., de Wolf, R.: Optimal parallel quantum query algorithms. Algorithmica 79(2), 509–529 (2017)
17. Liu, Q., Zhandry, M.: Revisiting post-quantum fiat-shamir. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019. Lecture Notes in Computer Science, vol. 11693, pp. 326–355. Springer (2019)
18. Unruh, D.: Revocable quantum timed-release encryption. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014. Lecture Notes in Computer Science, vol. 8441, pp. 129–146. Springer (2014)
19. Zalka, C.: Grover's quantum searching algorithm is optimal. Phys. Rev. A 60, 2746–2751 (Oct 1999), https://link.aps.org/doi/10.1103/PhysRevA.60.2746
20. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019. Lecture Notes in Computer Science, vol. 11693, pp. 239–268. Springer (2019), https://doi.org/10.1007/978-3-030-26951-7_9