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

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

[1]Academia Sinica, Taiwan (kmchung@iis.sinica.edu.tw)
[2]CWI Cryptology Group and Leiden University, The Netherlands (serge.fehr@cwi.nl)
[3]National Chiao-Tung University, Taiwan (asd00012334.cs04@nctu.edu.tw)
[4]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). This technique has proven to be very powerful for reproving known lower bound results, but also for proving new results that seemed to be out of reach before. Despite being very useful, it is however still quite cumbersome to actually employ the compressed oracle technique.

To start off with, we offer a *concise* yet *mathematically rigorous* exposition of the compressed oracle technique. We adopt a more abstract view than other descriptions found in the literature, which allows us to keep the focus on the relevant aspects. Our exposition easily extends to the *parallel-query* QROM, where in each query-round the considered quantum oracle algorithm may make *several* queries to the QROM *in parallel*. This variant of the QROM allows for a more fine-grained query-complexity analysis of quantum oracle algorithms.

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, we show 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 (like the optimality of parallel Grover), but also obtaining new results (like the optimality of parallel BHT collision search). Our main application is to prove hardness of finding a q-chain, i.e., a sequence $x_0, x_1, \ldots, x_q$ with the property that $x_i = H(x_{i-1})$ for all $1 \leq i \leq q$, with fewer than q parallel queries.

The above problem of producing a hash chain is of fundamental importance in the context of *proofs of sequential work*. Indeed, as a concrete application of our new bound, we prove that the "Simple Proofs of Sequential Work" proposed by Cohen and Pietrzak remain secure against quantum attacks. Such a proof is not simply a matter of plugging in our new bound; the entire protocol needs to be analyzed in the light of a quantum attack, and substantial additional work is necessary. Thanks to our framework, this can now be done with purely classical reasoning.

1 Introduction

Background. The random oracle methodology [2] has proven to be a successful way to design very efficient cryptographic protocols and arguing them secure in a rigorous yet idealized manner. The considered idealization treats a cryptographic hash function $H : \{0,1\}^n \to \{0,1\}^m$ as an external *oracle* that the adversary needs to query on $x \in \{0,1\}^n$ in order to learn $H(x)$. Furthermore, this oracle, called *random oracle* (RO) then, answers these queries by means of a *uniformly random* function $H : \{0,1\}^n \to \{0,1\}^m$. 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 reasonable 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 [14]. 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 gives rise to new and shorter proofs for known lower bound results on the query complexity of quantum algorithms (like Grover [3][10]); on the other hand, it allows for proving new cryptographic security results that seemed to be out of reach before, like in the context of *indifferentiability* [14], or, more recently, the *Fiat-Shamir transformation* [11], when considering a quantum adversary. Despite being very useful, it is however still 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 found in the literature in that we adopt a more abstract view in terms of Fourier transform for arbitrary finite Abelian groups, i.e., by considering the range of H to be an arbitrary finite Abelian group. Some readers may, to start with, feel uncomfortable with this approach, but it allows us to keep the focus on the relevant aspects, and, on the long run, abstraction simplifies matters and improves the understanding.

We also consider a generalization of the compressed-oracle technique to the *parallel-query* QROM. In this variation of the standard QROM, 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*, i.e., the queries are independent of the hash values that are to be learned. 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 whenever it is applicable, it is possible to prove *quantum* query complexity lower bounds by means of purely *classical* reasoning: all the quantum aspects are abstracted away by our framework. This means that no knowledge about quantum information science is actually 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 query complexity 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 $O(Q^2/2^m)$ 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 in parallel) is optimal [13]. 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 $O(Q^3/2^m)$ 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 [1][6], which makes k-parallel queries in the first phase to collect $kq/2$ function values and then runs a parallel Grover in the second phase, which gives a factor k^2 improvement. We are not aware of any prior optimality result on parallel collision search. 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 the property that that $x_i = H(x_{i-1})$ for all $1 \leq i \leq q$ (or, more generally, that $H(x_{i-1})$ is a substring of x_i, or yet satisfies some other relation). 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).[1] Here, we show that the same does hold in the quantum setting. Formally, we prove 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 [8] 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 than that: the entire protocol needs to be analyzed in the light of a quantum attack, and substantial additional work is necessary to reduce the security of the protocol to the hardness of finding a chain. As a matter of fact, we enrich our framework with a "calculus" that facilitates the latter. In return, relying on our framework, the proof of the quantum security of the PoSW scheme is purely classical, with no need to understand anything on quantum information science.

Recent Related Work. Independently and (partly) concurrently to the preparation of our work, the q-chain problem has been analyzed and tackled by another work by Blocki, Lee and Zhou [4], which we want to briefly discuss here.

As we do, Blocki, Lee and Zhou show the hardness of finding a q-chain with fewer than q queries for any quantum algorithm. Comparing the obtained bounds, we observe that our bound is significantly better. Translated into our notation, the upper bound on the success probability derived by Blocki, Lee and Zhou is $O(q\sqrt{k^3q^3/2^m})$, which is worse by more than a square-root compared to our bound $O(k^3q^3/2^m)$.[2] In addition, while Blocki *et al.* well emphasize the relevance of the q-chain problem to the PoSW by Cohen and Pietrzak [8], they do not offer an analysis of the latter.

We would also like to stress the conceptual differences in the respective contributions. In our work, we provide a general framework for proving quantum query complexity bounds by means of purely classical reasoning. With the framework in place, this makes our proof for the q-chain problem accessible to a much broader audience, and opens the door for non-quantum-experts to derive quantum query complexity bounds for their problems of interest. In contrast, Blocki *et al.*'s work is specific to the q-chain problem, and verifying the proof requires to go through cumbersome derivations that require a deep understanding of quantum information science in general and of the compressed oracle technique in particular.

2 Warm-up: Proving Classical Query Complexity Lower 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 as a reminder of how such classical bounds are (or can be) rigorously proven. Additionally, it demonstrates 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, let us briefly recall the *lazy sampling* technique, which allows to efficiently simulate the random oracle. Instead of choosing a uniformly random function $H : \mathcal{X} \to \mathcal{Y}$ and answering each query x to random oracle as $y = H(x)$, one can build up the hash function H "on the fly". Introduce a special symbol $\perp$, which stands for "not defined (yet)", 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 $x_i = x_j$ for some $j < i$. If this is the case then set $D_i = D_{i-1}$; else, do the following: 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 the latter is defined by

[1] The problem of finding a q-chain looks very similar to the *iterated hashing* studied by Unruh in [12]; however, a crucial difference is that the start of the chain, x_0, can be freely chosen here.

[2] We compare here the respective bounds under the strict requirement $H(x_{i-1}) = x_i$ for all i. Blocki *et al.* actually consider a notion of a q-chain (which they call an $\mathcal{H}$-sequence) where $H(x_{i-1})$ is asked to be a *continuous substring* of x_i, while we consider an *arbitrary relation* between $H(x_{i-1})$ and x_i. Our bound compares similarly favorably also for these variations.

$D_{i-1}[x_i \mapsto y_i](x_i) = y_i$ and $D_{i-1}[x_i \mapsto y_i](\bar{x}) = D_{i-1}(\bar{x})$ for $\bar{x} \neq x_i$. 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 random oracle; any algorithm making queries to the standard random oracle (which samples H 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 Efficient Representation

One important feature of the lazy-sampling technique is that it allows for an *efficient* simulation of the random oracle. Indeed, compared to a uniformly random function $H : \mathcal{X} \to \mathcal{Y}$, the databases $D_0, D_1, \ldots$ can be efficiently represented by means of an encoding function enc that maps any database $D : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ to (a suitable representation of) the list of pairs $\big(x, D(x)\big)$ for which $D(x) \neq \perp$.[3] Obviously, for a bounded number of queries, the list $enc(D_i)$ remains bounded in size. Furthermore, the update $enc(D_i) \mapsto enc(D_{i+1}) = enc(D_i[x_i \mapsto y_i])$ can be efficiently computed (for any choice of y_i).

2.3 Proving Classical Lower Bounds

In the work here, we are more interested in the fact that the lazy sampling idea is useful for showing lower bounds on the query complexity for certain tasks. Our goal here is to show on a few examples 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 the random oracle, 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 4.2 (originally by Zhandry).

The above observation implies that it is sufficient to show that $P[\exists\, x : D_q(x) = 0]$ is small. Furthermore, writing $\mathsf{PRMG} := \{D : \mathcal{X} \to \mathcal{Y} \cup \{\perp\} \mid \exists\, x : D(x) = 0\}$, we can write and decompose

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

In order to align the reasoning here with our framework, which relies on the notion of a *quantum transition capacity*, we introduce here 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 5.6 (with P_s set to PRMG), which is in terms of the (appropriately defined) *quantum* transition capacity $\big[\!\big[\cdot \to \cdot\big]\!\big]$.

[3] This representation as a list of pairs somewhat justifies the terminology "database" for D.

The reader probably already sees that $\big[\,\neg\mathsf{PRMG} \to \mathsf{PRMG}\,\big] = 1/|\mathcal{Y}|$, leading to the (well-known) bound $p \le (q+1)/|\mathcal{Y}|$. 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, $P[D^{\circlearrowleft x} \in \mathsf{PRMG}\,] = P[U \in \mathsf{L}^{D,x}]$ where U is defined to be uniformly random in $\mathcal{Y}$. Here, we can simply choose $\mathsf{L}^{D,x} := \{0\}$ and thus indeed 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 $\big[\!\big[\cdot \to \cdot\big]\!\big]$ 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 5.15, we obtain

$$\big[\!\big[\neg\mathsf{PRMG} \to \mathsf{PRMG}\big]\!\big] \le \max_{D,x} \sqrt{10P\big[U \in \mathsf{L}^{D,x}\big]} \le \sqrt{\frac{10}{|\mathcal{Y}|}}\,.$$

By Theorem 5.6, this implies that the success probability p of a *quantum* algorithm to find a preimage is bounded by

$$p \le \left(q\big[\!\big[\neg\mathsf{PRMG} \to \mathsf{PRMG}\big]\!\big] + \frac{1}{\sqrt{|\mathcal{Y}|}}\right)^2 \le \left(q\sqrt{\frac{10}{|\mathcal{Y}|}} + \frac{1}{\sqrt{|\mathcal{Y}|}}\right)^2 = O\Big(\frac{q^2}{|\mathcal{Y}|}\Big)\,,$$

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 random oracle, 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 $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$, we then identify a *family* of k 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, $P[D^{\circlearrowleft \mathbf{x}} \in \mathsf{PRMG}\,] \le \sum_i P[U \in \mathsf{L}_i^{D,x}]$. Setting $\mathsf{L}_1^{D,\mathbf{x}} = \ldots = \mathsf{L}_k^{D,\mathbf{x}} := \{0\}$, we now obtain $\big[\,\neg\mathsf{PRMG} \xrightarrow{k} \mathsf{PRMG}\,\big] = kP[U=0] = k/|\mathcal{Y}|$, showing a factor-$k$ increase in the bound as expected. More interesting is that Theorem 5.15 still applies, implying that for the quantum version we have

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

Plugging this into Theorem 5.6, we then get the bound

$$p \le \left(q\sqrt{\frac{10k}{|\mathcal{Y}|}} + \frac{1}{\sqrt{|\mathcal{Y}|}}\right)^2 = O\Big(\frac{q^2 k}{|\mathcal{Y}|}\Big)\,,$$

showing optimality of running k parallel executions of Grover.

Finding a Chain (with Parallel Queries). The other 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 $H(x_{i-1}) \lhd x_i$, with no more than q (parallel) queries. Here, $\lhd$ refers to an arbitrary relation among the elements of $\mathcal{X}$ and $\mathcal{Y}$; typical examples are: $y \lhd x$ if $x = y$, or if y is a prefix of y, or if y is an arbitrary continuous substring of x. Below, we set $\mathcal{Y}^{\lhd x} := \{y \in \mathcal{Y} \,|\, y \lhd x\}$ and $T := \max_x |\mathcal{Y}^{\lhd x}|$.

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

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

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, it will be useful to exploit that after s (parallel) queries, $D_s \in \mathsf{SZ}_{\leq ks} := \{D \,|\, |\{x|D(x) \neq \perp\}| \leq ks\}$, i.e., that the *size* of the database D_s, measured as the number of x's for which $D_s(x) \neq \perp$, is at most ks . Thus, the above extends to

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

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 (classical and quantum) 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 $i = 1, \ldots, k$:

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

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

Lemma 2.1. $D[\mathbf{x} \mapsto \mathbf{r}] \notin \mathsf{CHN}^s \;\wedge\; D[\mathbf{x} \mapsto \mathbf{u}] \in \mathsf{CHN}^{s+1} \implies \exists\, i : r_i \neq u_i \;\wedge\; u_i \in \mathsf{L}_i^{D,\mathbf{x}}$.

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 $D_\circ \in \mathsf{CHN}^{s+1}$ 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 $\neq \perp$. The latter means that either $D(\hat{x}_{s_\circ+1}) \neq \perp$ 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 $\{x|D(x) \neq \perp\}$ is bounded in size by $k(s-1)$, and $|\mathcal{Y}^{\triangleleft x}|, |\mathcal{Y}^{\triangleleft x_j}| \leq T$, 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 [8].

Also here, our framework allows us to lift the above reasoning to the quantum setting, simply 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 2.1 ensures that we can apply Theorem 5.19 to bound the *quantum* transition capacity as

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

where e is Euler's number. Plugging this into Theorem 5.6, we then get the bound

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

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 $y \triangleleft x$; $T = 1$ 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.

Finding a Collision (with Parallel Queries). In the same spirit, for the query complexity of finding a *collision*, it is sufficient to control the transition capacity for $\mathsf{CL} := \{D : \mathcal{X} \to \mathcal{Y} \cup \{\bot\} \,|\, \exists\, x \neq x' : \bot \neq D(x) = D(x') \neq \bot\}$. Indeed, using the same kind of reasoning as above, we can argue that

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

with the (classical) transition capacity here given by $\max P[D^{\circlearrowleft \mathbf{x}} \in \mathsf{CL}\,]$, maximized over all $D \in \mathsf{SZ}_{\leq k(s-1)} \setminus \mathsf{CL}$ and $\mathbf{x} \in \mathcal{X}^k$. In order to analyze this transition capacity, for any D and any $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$, we consider the following family of 2-local properties:

$$\mathsf{CL}_{i,j} = \{(y, y) \,|\, y \in \mathcal{Y}\} \qquad \text{and} \qquad \mathsf{CL}_i = \{D(\bar{x}) \,|\, \bar{x} \notin \{x_1, \ldots, x_k\} : D(\bar{x}) \neq \bot\}$$

for $i \neq j \in \{1, \ldots, k\}$, were we leave the dependency on D and $\mathbf{x}$ implicit. Similar to (2), here we have that for any $\mathbf{x} \in \mathcal{X}^k$ and $D \in \mathsf{SZ}_{\leq k(s-1)} \setminus \mathsf{CL}$

$$D[\mathbf{x} \mapsto \mathbf{y}] \in \mathsf{CL} \iff \big(\exists\, i \neq j : (y_i, y_j) \in \mathsf{CL}_{i,j}\big) \vee \big(\exists\, i : y_i \in \mathsf{CL}_i\big)\,,$$

i.e., a collision can only happen for $D[\mathbf{x} \mapsto \mathbf{y}]$ if $y_i = y_j$ for $i \neq j$, or $y_i = D(\bar{x})$ for some i and some $\bar{x}$ outside of $\mathbf{x}$. It then follows that

$$\left[\, \mathsf{SZ}_{\leq k(s-1)} \backslash \mathsf{CL} \xrightarrow{k} \mathsf{CL} \,\right] = \sum_{i \neq j} P[(U, U') \in \mathsf{CL}_{i,j}] + \sum_{i} P[U \in \mathsf{CL}_i] \leq \frac{k(k-1)}{|\mathcal{Y}|} + \frac{k^2(s-1)}{|\mathcal{Y}|}\,,$$

where we exploited that CL_i is bounded in size by assumption on D. This then amounts to the classical bound

$$p \leq O\!\left(\frac{q^2 k^2}{|\mathcal{Y}|}\right)$$

on the success probability of finding a collision with no more than q k-parallel queries.

Here, due to the 2-locality of $\mathsf{CL}_{i,j}$, there is an additional small complication for deriving the corresponding quantum bound, since in such a case our frameword does not relate the corresponding quantum transition capacity to the probability $P[(U, U') \in \mathsf{CL}_{i,j}]$ of a random pair in $\mathcal{Y} \times \mathcal{Y}$ satisfying the 2-local property $\mathsf{CL}_{i,j}$. Instead, we have to consider the following derived 1-local properties. For any $i \neq j$ and D', let

$$\mathsf{CL}_{i,j}|_{D'|^{x_i}} := \mathsf{CL}_{i,j} \cap \big((\mathcal{Y} \cup \{\bot\}) \times \{D'(x_j)\}\big) = \{D'(x_j)\} \qquad \text{and} \qquad \mathsf{CL}_i|_{D'|^{x_i}} := \mathsf{CL}_i\,.$$

Then, the considered quantum transition capacity is given in terms of

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

Namely, by Theorem 5.21,

$$\left[\!\!\left[\, \mathsf{SZ}_{\leq ks} \backslash \mathsf{CL} \xrightarrow{k} \mathsf{CL} \,\right]\!\!\right] \leq e2\sqrt{10\Big(\sum_{i \neq j} P\big[U \in \mathsf{CL}_{i,j}|_{D'|^{x_i}}\big] + \sum_{i} P\big[U \in \mathsf{CL}_i|_{D'|^{x_i}}\big]\Big)} \leq 2ek\sqrt{10\,\frac{q+1}{M}}\,.$$

By Theorem 5.6, this then amounts to the bound

$$p \leq O\!\left(\frac{q^3 k^2}{|\mathcal{Y}|}\right)$$

on the success probability of a quantum oracle algorithm in finding a collision with no more than q k-parallel queries.

3 Notation

3.1 Operators and Their Norms

Let $\mathcal{H}$ be a finite-dimensional complex Hilbert space; by default, $\mathcal{H} = \mathbb{C}^d$ for some dimension d. We use 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

$$\|A\| = \max_{1 \leq i \leq m} \|B_i\| \,.$$

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 consists of all group homomorphisms $\mathcal{Y} \to \mathbb{C}^\times = \{\omega \in \mathbb{C} \,|\, |\omega| = 1\}$ and is known 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 as groups 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.[4] With the above convention on the notation, we have $\mathbb{C}[\mathcal{Y}] = \mathbb{C}[\hat{\mathcal{Y}}] = \mathcal{H}$.[5] 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 $\mathbb{C}[\mathcal{Y} \cup \{\perp\}] = \mathbb{C}[\hat{\mathcal{Y}} \cup \{\perp\}]$.

[4] By fixing an isomorphism $\mathcal{Y} \to \hat{\mathcal{Y}}, y \mapsto \hat{y}$ we obtain a unitary map $|y\rangle \mapsto |\hat{y}\rangle$, called *quantum Fourier transform (QFT)*. However, we point out that *in general* there is no *natural* choice for the isomorphism, and thus for the QFT — but in the common cases there is. We note that in this work we do not fix any such isomorphism and do not make use of a QFT; we merely consider the two bases.

[5] The reader that feels uncomfortable with this abstract approach to the Fourier basis may stick to $\mathcal{Y} = \{0,1\}^m$ and replace $|\hat{y}\rangle$ by $\mathsf{H}^{\otimes m}|y\rangle$ with $y \in \{0,1\}^m$ and H the Hadamard matrix.

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 all 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 *labelled* 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, $|\hat{H}\rangle = \bigotimes_x |\hat{H}(x)\rangle$ is the "quantum representation" of $\hat{H} \in \hat{\mathfrak{H}}$. By applying (5) register-wise, any $|H\rangle$ decomposes into a linear combination of 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, the "quantum representation" of a database D is given by $|D\rangle = \bigotimes_x |D(x)\rangle \in \bigotimes_x \mathbb{C}[\bar{\mathcal{Y}}] = \mathbb{C}[\mathfrak{D}]$. 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 $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$, we write $D(\mathbf{x})$ for $\big(D(x_1), \ldots, D(x_k)\big) \in \bar{\mathcal{Y}}^k$; similarly for $H \in \mathfrak{H}$. Furthermore, if $\mathbf{x}$ has pairwise distinct entries and $\mathbf{r} = (r_1, \ldots, r_k) \in \bar{\mathcal{Y}}^k$, we define $D[\mathbf{x}\mapsto\mathbf{r}] \in \mathfrak{D}$ to be the database

$$D[\mathbf{x}\mapsto\mathbf{r}](x_i) = r_i \qquad \text{and} \qquad D[\mathbf{x}\mapsto\mathbf{r}](\bar{x}) = D(\bar{x}) \;\; \forall\, \bar{x} \notin \{x_1, \ldots, x_k\}\,.$$

4 Zhandry's Compressed Oracle - Refurbished

We give a concise yet self-contained and mathematically rigorous introduction to the compressed-oracle technique. For the reader familiar with the compressed oracle, we still recommend to browse over the section to familiarize with the notation we are using, and for some important observations, but some of the proofs can well be skipped then.

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 for any (quantum) query algorithm since the queries commute with measuring the superposition. 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, it follows that 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 register x for all $x \in \mathcal{X}$ (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 $\hat{H}(x) = 0$. As a consequence, the internal state of the compressed oracle after q queries is supported by state 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 is referred to as the *compressed* oracle because, for a bounded number of queries, these state vectors $|D\rangle$ can be efficiently represented in terms of the number of qubits, i.e., can be *compressed*, as $|enc(D)\rangle$, i.e., by employing a classical efficient representation, similar to the one mentioned in Section 2.2. Furthermore, the unitary that implements an oracle call (see cO below) can then be efficiently computed by a quantum circuit. In this work, we are not concerned with such computational efficiency aspect; nevertheless, for completeness, we formally discuss this in Appendix A.

4.2 Linking the Compressed and the Original Oracle

The following result (originally by Zhandry [14]) links the compressed oracle with the original standard oracle. Intuitively, it ensures that one can extract useful information from the compressed oracle.

Lemma 4.1. *Consider an arbitrary (normalized) $|\Pi\rangle \in \mathbb{C}[\mathfrak{H}]$, and 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 somewhat technical statement directly translates to the following statement in terms of algorithmic language.

Corollary 4.2 (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 uniformly 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 (in the computational basis). Then*

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

Proof (of Corollary 4.2). Consider an execution of $\mathcal{A}$ when interacting with the purified oracle. For technical reasons, we assume that, after having measured and output $\mathbf{x}, \mathbf{y}$, $\mathcal{A}$ measures its internal state in the computational basis to obtain a string w, which he outputs as well. We first observe that

$$p = \sum_{\substack{\mathbf{x},\mathbf{y},w \\ (\mathbf{x},\mathbf{y})\in R}} q_{\mathbf{x},\mathbf{y},w}\, p_{\mathbf{x},\mathbf{y},w} \qquad \text{and} \qquad p' = \sum_{\substack{\mathbf{x},\mathbf{y},w \\ (\mathbf{x},\mathbf{y})\in R}} q_{\mathbf{x},\mathbf{y},w}\, p'_{\mathbf{x},\mathbf{y},w}$$

where $q_{\mathbf{x},\mathbf{y},w}$ is the probability that $\mathcal{A}$ outputs the triple $\mathbf{x}, \mathbf{y}, w$, and $p_{\mathbf{x},\mathbf{y},w}$ is the probability that $\mathbf{y} = H(\mathbf{x})$ *conditioned* on the considered output of $\mathcal{A}$, and correspondingly for $p'_{\mathbf{x},\mathbf{y},w}$. More technically, using the notation from Lemma 4.1, $p_{\mathbf{x},\mathbf{y},w} = \|P_\mathbf{x}|\Pi\rangle\|^2$ with $|\Pi\rangle$ the internal state of the purified oracle, *post-selected* on $\mathbf{x}, \mathbf{y}$ and w. Similarly, $p'_{\mathbf{x},\mathbf{y},w} = \|P_\mathbf{x}\mathsf{Comp}\,|\Pi\rangle\|^2$. Thus, applying Lemma 4.1 and squaring, we obtain

$$p_{\mathbf{x},\mathbf{y},w} \leq \left(\sqrt{p'_{\mathbf{x},\mathbf{y},w}} + \varepsilon\right)^2 = p'_{\mathbf{x},\mathbf{y},w} + 2\sqrt{p'_{\mathbf{x},\mathbf{y},w}}\,\varepsilon + \varepsilon^2\,.$$

Averaging with the $q_{\mathbf{x},\mathbf{y},w}$'s, applying Jensen's inequality, and taking square-roots, then implies the claim. □

Proof (of Lemma 4.1). We set $\mathsf{Comp}_{\mathbf{x}} := \bigotimes_i \mathsf{Comp}_{x_i}$; the subscript $\mathbf{x}$ again emphasizing that $\mathsf{Comp}_{\mathbf{x}}$ acts on the registers $x_1, \dots, x_\ell$ only. In line with this, we write $\mathsf{I}_{\bar{\mathbf{x}}}$ for the identity acting on the registers $x \notin \{x_1, \dots, x_\ell\}$. Then[6]

$$\begin{aligned}
\|P_{\mathbf{x}}|\Pi\rangle\| - \|P_{\mathbf{x}}\mathsf{Comp}|\Pi\rangle\| &= \|P_{\mathbf{x}}|\Pi\rangle\| - \|P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}}|\Pi\rangle\| && \text{(since the } \mathsf{Comp}_x\text{'s are isometries)}\\
&\leq \|(P_{\mathbf{x}} - P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}})|\Pi\rangle\| && \text{(by triangle inequality)}\\
&\leq \|(P_{\mathbf{x}} - P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}}) \otimes \mathsf{I}_{\bar{\mathbf{x}}}\| && \text{(by definition of the operator norm)}\\
&= \|P_{\mathbf{x}} - P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}}\| && \text{(by basic property of the operator norm)}
\end{aligned}$$

We will work out the above operator norm. For this, recall that in the Fourier basis

$$P_{\mathbf{x}} = \bigotimes_i \left(\frac{1}{M} \sum_{\substack{\hat{y}\in\hat{\mathcal{Y}}\\ \hat{z}\in\hat{\mathcal{Y}}}} \omega_{\hat{z}/\hat{y}}(y_i) |\hat{z}\rangle\langle\hat{y}| \right) \quad \text{and} \quad \mathsf{Comp}_{\mathbf{x}} = \bigotimes_i \left(|\perp\rangle\langle 0| + \sum_{0\neq\hat{y}\in\hat{\mathcal{Y}}} |\hat{y}\rangle\langle\hat{y}| \right),$$

with the understanding that in the above respective tensor products the i-th component acts on register x_i, and where the $\omega_{\hat{z}/\hat{y}}(y_i)$ are suitable phases, i.e., norm-1 scalars, which will be irrelevant though.[7] By multiplying the two, we get

$$P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}} = \bigotimes_i \left(\frac{1}{M} \sum_{\substack{0\neq\hat{y}\in\hat{\mathcal{Y}}\\ \hat{z}\in\hat{\mathcal{Y}}}} \omega_{\hat{z}/\hat{y}}(y_i) |\hat{z}\rangle\langle\hat{y}| \right).$$

Multiplying out the respective tensor products in $P_{\mathbf{x}}$ and $P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}}$, and subtracting the two expressions, we obtain

$$P_{\mathbf{x}} - P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}} = \frac{1}{M^\ell} \sum_{\substack{\hat{y}_1,\dots,\hat{z}_\ell\in\hat{\mathcal{Y}}\\ \exists i:\hat{y}_i=0}} \bigotimes_i \omega_{\hat{z}_i/\hat{y}_i}(y_i) |\hat{z}_i\rangle\langle\hat{y}_i| = \frac{1}{M^\ell} \sum_{\substack{\hat{\mathbf{y}},\hat{\mathbf{z}}\\ \exists i:\hat{y}_i=0}} \omega_{\hat{\mathbf{z}}/\hat{\mathbf{y}}} |\hat{\mathbf{z}}\rangle\langle\hat{\mathbf{y}}|,$$

where the sum is over all $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_\ell)$ and $\hat{\mathbf{z}} = (\hat{z}_1, \dots, \hat{z}_\ell)$ in $\hat{\mathcal{Y}}^\ell$ subject to that at least one $\hat{y}_i$ is 0, and where $\omega_{\hat{\mathbf{z}}/\hat{\mathbf{y}}}$ is the phase $\omega_{\hat{\mathbf{z}}/\hat{\mathbf{y}}} := \prod_i \omega_{\hat{z}_i/\hat{y}_i}(y_i)$. Bounding the operator norm by the Frobenius norm, we thus obtain that

$$\begin{aligned}
\|P_{\mathbf{x}} - P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}}\|^2 &\leq \sum_{\hat{\mathbf{y}},\hat{\mathbf{z}}} |\langle\hat{\mathbf{z}}|(P_{\mathbf{x}} - P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}})|\hat{\mathbf{y}}\rangle|^2\\
&= \frac{1}{M^{2\ell}} \sum_{\substack{\hat{\mathbf{y}},\hat{\mathbf{z}}\\ \exists i:\hat{y}_i=0}} |\omega_{\hat{\mathbf{z}}/\hat{\mathbf{y}}}|^2 \leq \frac{1}{M^{2\ell}}\, \ell M^{2\ell-1} = \frac{\ell}{M},
\end{aligned}$$

where the inequality is a standard counting argument: there are ℓ choices for i, and for each i there are $M^{\ell-1}$ choices for $\hat{\mathbf{y}} \in \hat{\mathcal{Y}}^\ell$ with $\hat{y}_i = 0$ (however, $\hat{\mathbf{y}}$'s with multiple zeros are counted multiple times this way). □

4.3 Working Out the Transition Matrix

Here, we explicitly 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 will be convenient to extend the domain $\mathbb{C}[\mathcal{Y}]$ of $\mathsf{cO}_{x,\hat{y}}$ and of Comp_x to $\mathbb{C}[\bar{\mathcal{Y}}]$ by declaring that $\mathsf{cO}_{x,\hat{y}}|\perp\rangle = |\perp\rangle$ and $\mathsf{Comp}_x|\perp\rangle = |\hat{0}\rangle$. This turns $\mathsf{cO}_{x,\hat{y}}$ and Comp_x into unitaries on $\mathbb{C}[\bar{\mathcal{Y}}]$, and correspondingly then for O and Comp. We are now interested in

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

[6] In line with the discussion in Section 3.1, since it maps any $|\perp\rangle$-component to 0, $P_{\mathbf{x}}$ can be understood to have domain $\mathbb{C}[\mathcal{Y}]^{\otimes\ell}$ or $\mathbb{C}[\bar{\mathcal{Y}}]^{\otimes\ell}$; the same for its range. Thus, below, in $P_{\mathbf{x}}|\Pi\rangle$ it is understood as $\mathbb{C}[\mathcal{Y}]^{\otimes\ell} \to \mathbb{C}[\mathcal{Y}]^{\otimes\ell} \subseteq \mathbb{C}[\bar{\mathcal{Y}}]^{\otimes\ell}$, while in $P_{\mathbf{x}}\mathsf{Comp}_{\mathbf{x}}|\Pi\rangle$ as $\mathbb{C}[\bar{\mathcal{Y}}]^{\otimes\ell} \to \mathbb{C}[\bar{\mathcal{Y}}]^{\otimes\ell}$.

[7] For the record, switching back to multiplicative notation for the elements in the dual group $\hat{\mathcal{Y}}$, we have $\omega_{\hat{z}/\hat{y}}(y_i) = (\hat{z}/\hat{y})(y_i)$.

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 $\mathsf{cO}_{x\hat{y}} := \mathsf{Comp}_x \circ \mathsf{O}_{x\hat{y}} \circ \mathsf{Comp}_x^\dagger \in \mathcal{L}(\mathbb{C}[\bar{\mathcal{Y}}])$ acts on the x-register only. In the form of a commuting diagram, we thus have

$$\begin{array}{ccc} \mathbb{C}[\mathfrak{H}] & \xrightarrow{\mathsf{Comp}} & \mathbb{C}[\mathfrak{D}] \\ {\scriptstyle \mathsf{O}_{x\hat{y}}}\downarrow & & \downarrow{\scriptstyle \mathsf{cO}_{x\hat{y}}} \\ \mathbb{C}[\mathfrak{H}] & \xrightarrow{\mathsf{Comp}} & \mathbb{C}[\mathfrak{D}] \end{array}$$

Lemma 4.3. *For any $\hat{y} \neq 0$, in the computational basis the unitary $\mathsf{cO}_{x,\hat{y}}$ on $\mathbb{C}^{M+1}$ is represented by the matrix given in Figure 1; i.e, for all $r, u \in \bar{\mathcal{Y}} := \mathcal{Y} \cup \{\perp\}$ it holds that $\langle u|\mathsf{cO}_{x,\hat{y}}|r\rangle = \gamma^{\hat{y}}_{u,r}$. Furthermore, $\mathsf{cO}_{x,\hat{0}} = \mathsf{I}$.*

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

Figure 1: The matrix describing the evolution of the compressed oracle in the computational basis.

Proof. From simple but somewhat tedious manipulations, using basic properties of the Fourier transform, we obtain the following. For any $r \neq \perp$ (and $\hat{y} \neq \hat{0}$), we have

$$\sqrt{M}\,|r\rangle = \sum_{\hat{r}} \hat{r}(r)|\hat{r}\rangle = |0\rangle + \sum_{\hat{r}\neq\hat{0}} \hat{r}(r)|\hat{r}\rangle\,,$$

which gets mapped to

$$\xmapsto{\mathsf{Comp}^\dagger} |\perp\rangle + \sum_{\hat{r}\neq\hat{0}} \hat{r}(r)|\hat{r}\rangle\,,$$

which gets mapped to

$$\begin{aligned} \xmapsto{\mathsf{cO}_{x,\hat{y}}} |\perp\rangle + \sum_{\hat{r}\neq\hat{0}} \hat{r}(r)|\hat{r}+\hat{y}\rangle &= |\perp\rangle - |\hat{y}\rangle + \sum_{\hat{r}} \hat{r}(r)|\hat{r}+\hat{y}\rangle \\ &= |\perp\rangle - |\hat{y}\rangle + \hat{y}^*(r)\sum_{\hat{r}} \hat{r}(r)\,|\hat{r}\rangle = |\perp\rangle - |\hat{y}\rangle + \hat{y}^*(r)|\hat{0}\rangle + \hat{y}^*(r)\sum_{\hat{r}\neq\hat{0}} \hat{r}(r)\,|\hat{r}\rangle\,, \end{aligned}$$

which gets mapped to

$$\begin{aligned} \xmapsto{\mathsf{Comp}} |\hat{0}\rangle - |\hat{y}\rangle + \hat{y}^*(r)|\perp\rangle + \hat{y}^*(r)\sum_{\hat{r}\neq\hat{0}} \hat{r}(r)|\hat{r}\rangle & \\ = |\hat{0}\rangle - |\hat{y}\rangle + \hat{y}^*(r)|\perp\rangle - \hat{y}^*(r)|\hat{0}\rangle + \hat{y}^*(r)\sum_{\hat{r}} \hat{r}(r)\,|\hat{r}\rangle & \\ = \frac{1}{\sqrt{M}}\sum_u |u\rangle - \frac{1}{\sqrt{M}}\sum_u \hat{y}^*(u)|u\rangle + \hat{y}^*(r)|\perp\rangle - \frac{\hat{y}^*(r)}{\sqrt{M}}\sum_u |u\rangle + \sqrt{M}\hat{y}^*(r)|r\rangle\,. & \end{aligned}$$

From this expression, one can now easily read out the coefficients $\gamma^{\hat{y}}_{u,r}$ for $r \neq \perp$. Finally, from

$$|\perp\rangle \xmapsto{\mathsf{Comp}^\dagger} |\hat{0}\rangle \xmapsto{\mathsf{cO}_{x,\hat{y}}} |\hat{y}\rangle \xmapsto{\mathsf{Comp}} |\hat{y}\rangle = \frac{1}{\sqrt{M}} \sum_u \hat{y}^*(u)|u\rangle$$

we obtain the coefficients for $r = \perp$ (for$\hat{y} \neq 0$). The case $\hat{y} = 0$ follows from the fact that $\mathsf{O}_{x,\hat{0}} = \mathsf{I}$. □

Since, for any fixed $\hat{y}$, this matrix is unitary, the squares of the absolute values of each column add up to 1. Thus, for any $\hat{y}$ and r we can consider the (conditional) 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., $\perp \notin \mathsf{L}$),

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

where $P[U \in \mathsf{L}] = \frac{|\mathsf{L}|}{M}$ is the probability for a uniformly 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*. We recall that distinguishing between *parallel* and *sequential* queries allows for a more fine-grained query-complexity analysis of quantum algorithms. In particular, by showing a lower bound on the number of necessary *sequential* queries (with each sequential query possibly consisting of a large number of *parallel* queries), one can show the impossibility (or bound the possibility) of *parallelizing* computational tasks.

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 $\mathsf{cO}^k := \mathsf{Comp} \circ \mathsf{O}^k \circ \mathsf{Comp}^\dagger$, which described the evolution of the compressed oracle under such a k-parallel query, then acts as

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

for any $|\Delta\rangle \in \mathbb{C}[\mathfrak{D}]$, 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 (irrespectively of x_i and x_j being different or not).

5 A Framework for Proving Quantum Query Lower 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 5.1. *A* database property *on $\mathfrak{D}$ is a subset $\mathsf{P} \in \mathfrak{D}$ of the set of databases D.*

Remark 5.2. As the naming suggests, we think of P as a property that is either *true* or *false* for any $D \in \mathfrak{D}$; we thus also write $\mathsf{P}(D)$ to denote that $D \in \mathsf{P}$, i.e., to express that "D satisfies P". Furthermore, by convention, for any database property $\mathsf{P} \in \mathfrak{D}$, we overload notation and use P also to refer to the projection $\sum_{D\in\mathsf{P}} |D\rangle\langle D| \in \mathcal{L}(\mathbb{C}[\mathfrak{D}])$.

Examples that we will later consider are

$$\mathsf{PRMG} := \{D \,|\, \exists\, x : D(x) = 0\} \quad \text{and} \quad \mathsf{CL} = \{D \,|\, \exists\, x, x' : D(x) = D(x') \neq \bot\}$$

as well as

$$\mathsf{CHN}^q = \{D \,|\, \exists\, x_0, x_1, \ldots, x_q \in \mathcal{X} : D(x_{i-1}) \triangleleft x_i \;\forall i\}\,,$$

where $\triangleleft$ denotes an arbitrary relation, e.g., $y \triangleleft x$ 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 $D|^{\mathbf{x}}$.[8]

Remark 5.3. For fixed choices of $\mathbf{x}$ and D, we can, and often will, identify $D|^{\mathbf{x}}$ with $\bar{\mathcal{Y}}^k$ by means of the obvious identification 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

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

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

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

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.

Lemma 5.4. *For any two properties* P *and* P'*, and for any state* $|\phi\rangle$*,*

$$\|\mathsf{P}'\mathsf{cO}\,|\phi\rangle\| \leq \|\mathsf{P}|\phi\rangle\| + \max_{\mathbf{x},\hat{\mathbf{y}}} \|\mathsf{P}'\mathsf{cO}_{\mathbf{x},\hat{\mathbf{y}}}\,(\mathsf{I} - \mathsf{P})\| \leq \|\mathsf{P}|\phi\rangle\| + \max_{\mathbf{x},\hat{\mathbf{y}},D} \|\mathsf{P}'|_{D|^{\mathbf{x}}}\,\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\,(\mathsf{I} - \mathsf{P}|_{D|^{\mathbf{x}}})\|\,.$$

Proof. First, we see that

$$\|\mathsf{P}'\mathsf{cO}\,|\phi\rangle\| \leq \|\mathsf{P}'\mathsf{cO}\,\mathsf{P}|\phi\rangle\| + \|\mathsf{P}'\mathsf{cO}\,(\mathsf{I} - \mathsf{P})|\phi\rangle\| \leq \|\mathsf{P}|\phi\rangle\| + \|\mathsf{P}'\mathsf{cO}\,(\mathsf{I} - \mathsf{P})|\phi\rangle\|\,.$$

Then, we note that

$$\|\mathsf{P}'\mathsf{cO}\,(\mathsf{I} - P)|\phi\rangle\| \leq \|\mathsf{P}'\mathsf{cO}\,(\mathsf{I} - \mathsf{P})\| \leq \max_{\mathbf{x},\hat{\mathbf{y}}} \|\mathsf{P}'\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\,(\mathsf{I} - \mathsf{P})\|\,,$$

where the first inequality is by definition of the operator norm, and for the second we observe that $\mathsf{P}'\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\,(\mathsf{I} - \mathsf{P})$ maps $|\mathbf{x}\rangle|\hat{\mathbf{y}}\rangle \otimes |\Gamma\rangle$ to $|\mathbf{x}\rangle|\hat{\mathbf{y}}\rangle \otimes \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}|\Gamma\rangle$, and so the first inequality holds by basic properties of the operator norm.

For any fixed D, consider the subspace of $\mathfrak{D}$ 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 $\mathsf{I} - P$ and P'). Also, $\mathsf{cO}_{\mathbf{xy}}$ is a unitary on this subspace. The claim then again follows by basic properties of the operator norm. □

[8] We typically think of $\mathsf{P}|_{D|^{\mathbf{x}}}$ as a property of functions in D' in $D|^{\mathbf{x}}$.

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

Definition 5.5 (Quantum transition capacity)**.** *Let* P, P' *be two database properties. Then, the* quantum transition capacity *(of order* k*) is defined as*

$$\left[\!\left[\neg\mathsf{P} \xrightarrow{k} \mathsf{P}'\right]\!\right] := \max_{\mathbf{x},\hat{\mathbf{y}},D} \left\|\mathsf{P}'|_{D|^{\mathsf{x}}}\, \mathsf{cO}_{\mathbf{xy}}\, (\mathsf{I} - \mathsf{P}|_{D|^{\mathsf{x}}})\right\| .$$

More generally,

$$\left[\!\left[\neg\mathsf{P} \xrightarrow{k,q} \mathsf{P}'\right]\!\right] := \min_{\substack{\mathsf{P}_0,\ldots,\mathsf{P}_q \\ \mathsf{P}_0=\mathsf{P},\mathsf{P}_q=\mathsf{P}'}} \sum_{s=1}^{q} \left[\!\left[\neg\mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s\right]\!\right] .$$

The intuition behind the notation is that $\left[\!\left[\neg\mathsf{P} \to \mathsf{P}'\right]\!\right]$ represents a measure of how likely it is that, as a result of a query (or several queries), a compressed database $D \in \mathfrak{D}$ that does *not* satisfy P turns into a database D' that satisfies P'. We also use natural variations of this notation, like $\left[\!\left[\bot \to \mathsf{P}'\right]\!\right]$, which captures how likely it is that the initial all-$\bot$ database turns into a database that satisfies P', or $\left[\!\left[\mathsf{Q}\backslash\mathsf{P} \to \mathsf{P}'\right]\!\right]$, which captures how likely it is that the database that satisfies Q but not P turns into a database that satisfies P'. We also write $\neg\mathsf{P} \to \mathsf{P}'$ and refer to this as a *database transition* when considering two database properties P and P', and similarly with the above variations.

Theorem 5.6. *Let* R *be a relation, and let* $\mathcal{A}$ *a* k*-parallel* q*-query quantum oracle algorithm, both as in Corollary 4.2. Consider the database property*

$$\mathsf{P}^R = \left\{D \in \mathfrak{D} \mid \exists\, \mathbf{x} \in \mathcal{X}^\ell : \big(\mathbf{x}, H(\mathbf{x})\big) \in R\right\}$$

induced by R*. Then,* $\sqrt{p} \le \left[\!\left[\bot \xrightarrow{k,q} \mathsf{P}^R\right]\!\right] + \sqrt{\frac{\ell}{M}}$*, i.e.,*

$$\sqrt{p} \le \min_{\substack{\mathsf{P}_0,\ldots,\mathsf{P}_q \\ \mathsf{P}_0=\neg\bot,\mathsf{P}_q=\mathsf{P}^R}} \sum_{s=1}^{q} \left[\!\left[\neg\mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s\right]\!\right] + \sqrt{\frac{\ell}{M}} .$$

Remark 5.7. 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 $\left[\!\left[\neg\mathsf{P}_{s-1} \to \mathsf{P}_s\right]\!\right]$ are small. Often, it is good to keep track of the (growing but bounded) size of the database and instead bound the capacities

$$\left[\!\left[\mathsf{SZ}_{\le k(s-1)}\backslash\mathsf{P}_{s-1} \to \mathsf{P}_s\right]\!\right] = \left[\!\left[\mathsf{SZ}_{\le k(s-1)}\backslash\mathsf{P}_{s-1} \to \mathsf{P}_s \cup \neg\mathsf{SZ}_{\le ks}\right]\!\right] ,$$

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 instantiate the $\min$ in the definition of $\left[\!\left[\bot \xrightarrow{k,q} \mathsf{P}^R\right]\!\right]$ with $\mathsf{P}'_s = \neg(\mathsf{SZ}_{\le ks} \setminus \mathsf{P}_s) = \mathsf{P}_s \cup \neg\mathsf{SZ}_{\le ks}$.

Proof (of Theorem 5.6). Let $\mathsf{P}_0, \ldots, \mathsf{P}_q$ achieve the minimum in the definition of $\left[\!\left[\bot \xrightarrow{k,q} \mathsf{P}^R\right]\!\right]$. Applying Lemma 5.4 to $|\phi\rangle = |\phi_q\rangle$, the state produced by $\mathcal{A}$ after q queries, and to $\mathsf{P}' = \mathsf{P}_q$ and $\mathsf{P} = \mathsf{P}_{q-1}$, and applying induction (and using, for the base case, that $\|\mathsf{P}_0|\phi_0\rangle\| = 0$), we obtain that

$$\|\mathsf{P}_q|\phi_q\rangle\| \le \sum_{i=1}^{q} \left[\!\left[\neg\mathcal{G}_{i-1} \xrightarrow{k} \mathcal{G}_i\right]\!\right] + \sqrt{\frac{\ell}{M}} .$$

Further note that $\|\mathsf{P}_q|\phi_q\rangle\| = \|\mathsf{P}^R|\phi_q\rangle\|$, which equals the square-root of the probability that there exists $\mathbf{x}$ so that $(\mathbf{x}, D(\mathbf{x})) \in R$, with D obtained as in Corollary 4.2. Thus, the same upper bound applies to the probability that $\mathbf{x}$ output by $\mathcal{A}$ satisfies $(\mathbf{x}, D(\mathbf{x})) \in R$. The claim then follows from Corollary 4.2. □

In the following section, we offer techniques to bound the quantum transition capacities (in certain cases) using *purely classical* reasoning. In connection with Theorem 5.6, this then allows to prove lower bounds on the quantum query complexity (for certain computational problem in the random oracle model) using purely classical reasoning.

5.2 Bounding Quantum Transition Capacities Using Classical Reasoning Only

The general idea is to "recognize" a database transition $\neg\mathsf{P} \to \mathsf{P}$ in terms of *local* properties L, for which the truth value $\mathsf{L}(D)$ only depends on the function value $D(x)$ at *one single point* x (or at few points), and then to exploit that the behavior of the compressed oracle at a single point x is explicitly given by Lemma 4.3. In the following two sections, we consider two possible ways to do this, but first we provide the formal definition for local properties.

Definition 5.8. *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* $\mathsf{L}(D)$ *is uniquely determined by* $D(\mathbf{x})$*, and*
2. *if* $D \in \mathsf{L} \wedge \exists\, i \in \{1, \ldots, \ell\} : D(x_i) = \perp$ *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 5.9. 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.[9] 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 $\mathbf{x} = (x_1, \ldots, x_k)$.[10]

Remark 5.10. Condition 2 captures that $\perp$ is a dummy symbol with no more "value" than any other $r \in \mathcal{Y}$.

For example, for any database property P, and for any $\mathbf{x} = (x_1, \ldots, x_\ell)$ and D, the property $\mathsf{P}|_{D|^{\mathbf{x}}}$ satisfies requirement 1. of Definition 5.8. In line with this, Remark 5.3 applies here as well: we may identify an ℓ-local property L with a subset of $\bar{\mathcal{Y}}^\ell$.

5.2.1 Reasoning via Strong Recognizability

Definition 5.11. *A database transition* $\neg\mathsf{P} \to \mathsf{P}'$ *is said be* (uniformly) strongly recognizable *by* ℓ*-local properties if there exists a family of* ℓ*-local properties* $\{\mathsf{L}_i\}_i$ *so that*

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

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

Definition 5.12. *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* t*-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 5.13. 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* supports in Definition 5.11, and the same for the $\mathsf{L}_i^{\mathbf{x},D}$'s for every $\mathbf{x}$ and D in Definition 5.12.

Remark 5.14. It is easy to see that Definition 5.11 implies Definition 5.12 with $\mathsf{L}_i^{\mathbf{x},D} := \mathsf{L}_i|_{D|^{\mathbf{x}}}$.

[9] E.g., we may consider the constant-true property L with support $\mathsf{Supp}(\mathsf{L}) = \emptyset$, in which case it is ℓ-local for any $\ell \geq 0$, or we may consider the same constant-true property L but now with the support set to $\mathsf{Supp}(\mathsf{L}) = \{x_\circ\}$ for some $x_\circ \in \mathcal{X}$, which then is ℓ-local for $\ell \geq 1$.

[10] The above mentioned alternative approach would give $\subseteq$.

Theorem 5.15. *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, without loss of generality, the support of $\mathsf{L}_i^{\mathbf{x},D}$ is $\{x_i\}$. Then*

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

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

Before doing the proof, let us show how the above can be used to bound the probability of finding a 0-preimage.

Example. $\mathsf{P}' = \mathsf{P} = \mathsf{PRMG}$ is uniformly strongly recognized by the 1-local properties $\mathsf{L}_x = \{D | D(x) = 0\}$. Furthermore, 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 trivial.[11] In the non-trivial 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 5.15 that

$$[\![\neg\mathsf{PREIMG} \xrightarrow{k} \mathsf{PREIMG}]\!] \leq \sqrt{\frac{10k}{M}}\,,$$

and thus from Theorem 5.6, setting $\mathsf{P}_i = \mathsf{PREIMG}$ for all i, that the probability p of any k-parallel q-query algorithm outputting a 0-preimage x is bounded by

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

Proof (of Theorem 5.15). 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 $i \in \{1, \ldots, k\}$. By assumption (10), as projectors they satisfy

$$\mathsf{P}'|_{D|^{\mathbf{x}}} \leq \sum_i \mathsf{M}_i \leq \sum_i \mathsf{L}_i \qquad \text{and} \qquad \forall i : \mathsf{M}_i \leq \mathsf{L}_i \leq \mathsf{P}|_{D|^{\mathbf{x}}}\,,$$

where, additionally, the M_i's are mutually orthogonal. Then, exploiting the various properties, for any $\hat{\mathbf{y}}$ we have

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

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

$$\|\mathsf{L}_i\,\mathsf{cO}_{x_i\hat{y}_i}\,(\mathsf{I} - \mathsf{L}_i)\|^2 \leq \sum_{r_i,u_i \in \bar{\mathcal{Y}}} |\langle u_i|\mathsf{L}_i\,\mathsf{cO}_{x_i\hat{y}_i}\,(\mathsf{I} - \mathsf{L}_i)|r_i\rangle|^2 = \sum_{\substack{r_i \notin \mathsf{L}_i \\ u_i \in \mathsf{L}_i}} |\langle u_i|\mathsf{cO}_{x_i\hat{y}_i}|r_i\rangle|^2 = \sum_{r_i \notin \mathsf{L}_i} \tilde{P}[U \in \mathsf{L}_i | r_i, \hat{y}_i]\,.$$

The claim now follows from (8), with the additional observations that if $\perp \in \mathsf{L}_i$ (in which case (8) does not apply) then L_i is constant-true (by property 2 of Definition 5.8), and that the sum vanishes if L_i is constant-true. □

5.2.2 Reasoning via Weak Recognizability

Here, we consider a weaker notion of recognizability, which 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.

[11] 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 $D(x) \neq 0$.

Definition 5.16. *A database transition* $\mathsf{P} \to \mathsf{P}'$ *is said be* (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 5.13 and 5.14 apply correspondingly;[12] in particular, we may assume the supports in the considered families of local properties to be distinct.

Definition 5.17. *A database transition* $\mathsf{P} \to \mathsf{P}'$ *is said be* k-non-uniformly weakly recognizable *by* ℓ*-local properties if for every* $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{X}^k$ *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* $\{x_1, \ldots, x_k\}$ *so that*

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

Remark 5.18. Viewing $\mathsf{L}_i^{\mathbf{x},D}$ as subset of $\bar{\mathcal{Y}}^k$, and its support $\mathsf{L}_i^{\mathbf{x},D} = \{x_{i_1}, \ldots, x_{i_\ell}\}$ then as subset $\{i_1, \ldots, i_\ell\}$ of $\{1, \ldots, k\}$, (11) can equivalently be written as follows, which is in line with Lemma 2.1 (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. Consider $\mathsf{CHN}^q = \{D \mid \exists\, x_0, x_1, \ldots, x_q \in \mathcal{X} : D(x_{i-1}) \lhd x_i \,\forall i\}$ 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 $u \lhd x$ for some x with $D(x) \neq \bot$ or $x \in \{x_1, \ldots, x_k\}$. Lemma 2.1 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 $\mathsf{SZ}_{\leq k(q-1)} \setminus \mathsf{CHN}^q \to \mathsf{CHN}^{q+1}$; in this latter case however, whenever D is not in $\mathsf{SZ}_{\leq kq}$, which then means that the left hand side of (11) is never satisfied, we may simply pick the constant-false property as family of local properties satisfying (11).

Theorem 5.19. *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*

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

where e *is Euler's number.*[13]

Example. In the above example regarding CHN^q with the considered L_i's for $D \in \mathsf{SZ}_{\leq kq}$, as in the derivation of the classical bound in Section 2.3, it holds that $P[U \in \mathsf{L}_i] \leq kqT/M$, where T denotes the maximal number of $y \in \mathcal{Y}$ with $y \lhd x$ (for any x).[14] Thus,

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

and applying Theorem 5.6 (and the subsequent remark) to the database transitions $\mathsf{SZ}_{\leq k(s-1)} \setminus \mathsf{CHN}^s \to \mathsf{CHN}^{s+1}$ 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 5.20. *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*

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

where $T := \max_x |\{y \in \mathcal{Y} \mid y \lhd x\}|$*, and* M *is the size of the range* $\mathcal{Y}$ *of* $H : \mathcal{X} \to \mathcal{Y}$*.*

[12] We point out that this is thanks to our convention on the definition of the support, as discussed in Remark 5.9.

[13] Unlike Theorem 5.15, here is no convention that $P\big[U \in \mathsf{L}_i^{\mathbf{x},D}\big] = 0$ if $\mathsf{L}_i^{\mathbf{x},D}$ is constant-true. This has little relevance since $\mathsf{L}_i^{\mathbf{x},D}$ being constant-true can typically be avoided via Remark 5.13.

[14] For $D \notin \mathsf{SZ}_{\leq kq}$ we get the trivial bound 0 since we may then choose L_i to be constant false.

Proof (of Theorem 5.19). 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

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

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,[15]

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

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

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

Indeed, by definition of $\mathsf{P}'|_{D|^{\mathsf{x}}}$ and $\mathsf{P}|_{D|^{\mathsf{x}}}$ (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 $\mathbf{r} \in \mathsf{P}|_{D|^{\mathsf{x}}}$ and $\mathbf{u} \in \mathsf{P}'|_{D|^{\mathsf{x}}}$. 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

$$\|\mathsf{P}'|_{D|^{\mathsf{x}}}\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, \mathsf{P}|_{D|^{\mathsf{x}}}\| \leq \Bigg\| \sum_{i=0}^{k-1} \Big(\prod_{j<k-i} \mathsf{cO}_{x_j \hat{y}_j} \Big) A_{k-i} \Big(\prod_{j>k-i} B_j \Big) \Bigg\| \leq \sum_{i=0}^{k-1} \Big(\|A_{k-i}\| \prod_{j>k-i} \|B_j\| \Big)\,.$$

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

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

where 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. Using the fact that the operator norm is upper bounded by the Frobenius norm, we observe that

$$\|A_i\|^2 \leq \sum_{r_i, u_i} |\langle u_i|A_i|r_i\rangle|^2 = \sum_{\substack{u_i, r_i \text{ s.t.} \\ u_i \in L_i \wedge r_i \neq u_i}} |\langle u_i|\mathsf{cO}_{x_i y_i}|r_i\rangle|^2 = \sum_{r_i} \tilde{P}[r_i \neq U \in L_i | r_i, y_i] \leq 10 P[U \in \mathsf{L}_i]\,,$$

where the last inequality is 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 5.8, $\mathsf{L}_i = \bar{\mathcal{Y}}$, and hence the bound holds trivially. □

5.3 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.

[15] I.e., strictly speaking, we have $\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}} = \bigotimes_{j=1}^{k} \mathsf{cO}_{x_j \hat{y}_j}$.

Theorem 5.21. *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*

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

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

In case of *uniform* recognizability, where the is no dependency of L_t on $\mathbf{x}$ and D, the quantification over first D and then over $D' \in D|^{\mathsf{Supp}(\mathsf{L}_t)}$ collapses to a single quantification over D, simplifying the statement again a bit.

Corollary 5.22. *Let $\mathsf{P} \to \mathsf{P}'$ be a database transition that is uniformly strongly recognizable by ℓ-local properties L_t. Then*

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

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

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

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

for $i \neq j \in \{1, \ldots, k\}$, with respective supports $\{x_i, x_j\}$ and $\{x_i\}$. Note that these local properties depend on $\mathbf{x}$ but work uniformly for any choice of D.

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 $(\mathsf{SZ}_{\le ks} \setminus \mathsf{CL}) \to \mathsf{CL}$. In this case though, whenever $D \notin \mathsf{SZ}_{\le k(s+1)}$ 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 $\mathbf{x} = (x_1, \ldots, x_k)$ and $D \in \mathsf{SZ}_{\le k(s+1)}$ with $s \le q$. Then, for $i \neq j$, as subsets of $\bar{\mathcal{Y}}$ we have that

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

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

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

So, by Theorem 5.21,

$$\left[\!\left[\mathsf{SZ}_{\le ks} \setminus \mathsf{CL} \xrightarrow{k} \mathsf{CL}\right]\!\right] \le 2e\sqrt{10\left(\frac{k^2}{M} + \frac{k^2 q}{M}\right)} = 2ek\sqrt{10\,\frac{q+1}{M}}$$

and hence, by Theorem 5.6, we obtain the following bound.

Theorem 5.23. *The probability p of any k-parallel q-query algorithm outputting a collision is bounded by*

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

Proof (of Theorem 5.21) . We now consider an arbitrary but fixed choice of t and write L for L_t. We write $\{x_1, \ldots, x_\ell\}$ for its support and set $\mathbf{x}' := (x_1, \ldots, x_\ell)$. In order to control $\|\mathsf{L}\,\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\,(\mathsf{I}-\mathsf{L})\|$, we use a similar technique as in the proof of Theorem 5.19.[16]

For any $x_i \in \mathsf{Supp}(\mathsf{L})$, we set

$$A_i := \mathsf{L}\,\mathsf{cO}_{x_i\hat{y}_i}(\mathsf{I}-\mathsf{L}) \qquad \text{and} \qquad B_i := \mathsf{cO}_{x_i\hat{y}_i} - A_i\,.$$

By means of the same generic manipulations as in the proof of Theorem 5.19, we have

$$\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'} = \prod_{i=1}^{\ell} \mathsf{cO}_{x_i\hat{y}_i} = \sum_{i=0}^{\ell} \Big(\prod_{j<\ell-i} \mathsf{cO}_{x_j\hat{y}_j} \Big) A_{\ell-i} \Big(\prod_{j>\ell-i} B_j \Big)$$

with the convention that $A_0 = \mathsf{I}$. Furthermore, using $B_i(\mathsf{I}-\mathsf{L}_t) = (\mathsf{I}-\mathsf{L}_t)\mathsf{cO}_{x_{\lambda_i}\hat{y}_{\lambda_i}}(\mathsf{I}-\mathsf{L}_t)$, we see that

$$\mathsf{L}_t \Big(\prod_{j>0} B_j \Big) (\mathsf{I}-\mathsf{L}_t) = 0\,.$$

As a consequence, verbatim as in the proof of Theorem 5.19, we obtain

$$\|\mathsf{L}\,\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\,(\mathsf{I}-\mathsf{L})\| \le \sum_{i=0}^{\ell-1} \Big(\|A_{\ell-i}\| \prod_{j>\ell-i} \|B_j\| \Big) \le \sum_{i=1}^{\ell} \|A_i\|\, e\,.$$

Furthermore, for any $D' \in D|^{\mathbf{x}}$ and $i \in \{1, \ldots, \ell\}$, on the subspace spanned by $D'|^{x_i}$, the map A_i acts identically to $\mathsf{L}|_{D'|^{x_i}}\,\mathsf{cO}_{x_i\hat{y}_i}\,(\mathsf{I}-\mathsf{L}|_{D'|^{x_i}})$, and thus, by basic properties of the operator norm, the norm of A_i equals the largest norm of these restrictions:

$$\|A_i\| \le \big\| \mathsf{L}|_{D'|^{x_i}}\,\mathsf{cO}_{x_{\lambda_i}\hat{y}_i}\,(\mathsf{I}-\mathsf{L}|_{D'|^{x_i}}) \big\|\,.$$

Bounding the operator norm by the Frobenius norm, we then obtain

$$\|A_i\|^2 \le \sum_{\substack{r \notin L|_{D'|^{x_i}} \\ u \in L|_{D'|^{x_i}}}} |\langle u|\mathsf{cO}_{x_i\hat{y}_i}|r\rangle|^2 \le \sum_r \tilde{P}[r \neq U \in \mathsf{L}|_{D'|^{x_i}} | r, y_i] \le 10P\big[U \in \mathsf{L}|_{D'|^{x_i}}\big]\,.$$

where the last inequality is due to (8), with the additional observation that if $\perp \in \mathsf{L}_i$ then, by condition 2 of Definition 5.8, $\mathsf{L}|_{D'|^{x_i}} = \bar{\mathcal{Y}}$, and thus the sum vanishes. □

5.4 Some Rules for the Quantum Transition Capacity

As we have seen, certain "simple" lower bounds on the query complexity (respectively upper bound 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 show some useful manipulation rules here.

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:

$$\big[\!\big[\mathsf{P} \xrightarrow{k} \mathsf{P}' \big]\!\big] = \big[\!\big[\mathsf{P}' \xrightarrow{k} \mathsf{P} \big]\!\big]\,.$$

Therefore, the following bounds hold correspondingly also for $\big[\!\big[\mathsf{P} \xrightarrow{k} \mathsf{P}' \cap \mathsf{Q} \big]\!\big]$ etc.

Lemma 5.24. *For any database properties* P, P' *and* Q*,*

$$\big[\!\big[\mathsf{P} \cap \mathsf{Q} \xrightarrow{k} \mathsf{P}' \big]\!\big] \le \min\big\{ \big[\!\big[\mathsf{P} \xrightarrow{k} \mathsf{P}' \big]\!\big], \big[\!\big[\mathsf{Q} \xrightarrow{k} \mathsf{P}' \big]\!\big] \big\} \qquad \textit{and}$$

$$\max\big\{ \big[\!\big[\mathsf{P} \xrightarrow{k} \mathsf{P}' \big]\!\big], \big[\!\big[\mathsf{Q} \xrightarrow{k} \mathsf{P}' \big]\!\big] \big\} \le \big[\!\big[\mathsf{P} \cup \mathsf{Q} \xrightarrow{k} \mathsf{P}' \big]\!\big] \le \big[\!\big[\mathsf{P} \xrightarrow{k} \mathsf{P}' \big]\!\big] + \big[\!\big[\mathsf{Q} \xrightarrow{k} \mathsf{P}' \big]\!\big]\,.$$

[16] We point out that $\mathsf{L}(D')$ is determined by $D'(\mathbf{x}')$; thus, we may consider L as a property of functions $D' \in D|^{\mathbf{x}'} \subseteq D|^{\mathbf{x}}$.

In particular, we have the following intuitive rule.

Corollary 5.25. *If* $\mathsf{P} \subseteq \mathsf{Q}$ *then* $[\![\mathsf{P} \xrightarrow{k} \mathsf{P}']\!] \leq [\![\mathsf{Q} \xrightarrow{k} \mathsf{P}']\!]$ *and* $[\![\mathsf{P}' \xrightarrow{k} \mathsf{P}]\!] \leq [\![\mathsf{P}' \xrightarrow{k} \mathsf{Q}]\!]$.

Proof (of Lemma 5.24). As subsets, $(\mathsf{P} \cap \mathsf{Q})|_{D|^{\mathbf{x}}} = (\mathsf{P} \cap \mathsf{Q}) \cap D|^{\mathbf{x}} = (\mathsf{P} \cap D|^{\mathbf{x}}) \cap (\mathsf{Q} \cap D|^{\mathbf{x}}) = \mathsf{P}|_{D|^{\mathbf{x}}} \cap \mathsf{Q}|_{D|^{\mathbf{x}}}$, and, as projections, $\mathsf{P}|_{D|^{\mathbf{x}}}$ and $\mathsf{Q}|_{D|^{\mathbf{x}}}$ commute, and $\mathsf{P}|_{D|^{\mathbf{x}}} \cap \mathsf{Q}|_{D|^{\mathbf{x}}} = \mathsf{P}|_{D|^{\mathbf{x}}}\mathsf{Q}|_{D|^{\mathbf{x}}} = \mathsf{Q}|_{D|^{\mathbf{x}}}\mathsf{P}|_{D|^{\mathbf{x}}}\mathsf{Q}|_{D|^{\mathbf{x}}} \leq \mathsf{P}|_{D|^{\mathbf{x}}}$ and similarly $\leq \mathsf{Q}|_{D|^{\mathbf{x}}}$. This implies that

$$\|\mathsf{P}'|_{D|^{\mathbf{x}}}\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, (\mathsf{P} \cap \mathsf{Q})|_{D|^{\mathbf{x}}}\| \leq \min\left\{\|\mathsf{P}'|_{D|^{\mathbf{x}}}\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, \mathsf{P}|_{D|^{\mathbf{x}}}\|, \|\mathsf{P}'|_{D|^{\mathbf{x}}}\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, \mathsf{Q}|_{D|^{\mathbf{x}}}\|\right\},$$

and thus proves the first claim. Similarly, but now using that, as projections,

$$\mathsf{P}|_{D|^{\mathbf{x}}}, \mathsf{Q}|_{D|^{\mathbf{x}}} \leq \mathsf{P}|_{D|^{\mathbf{x}}} \cup \mathsf{Q}|_{D|^{\mathbf{x}}} \leq \mathsf{P}|_{D|^{\mathbf{x}}} + \mathsf{Q}|_{D|^{\mathbf{x}}},$$

we obtain the second claim. □

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 $X \subseteq \mathcal{X}$.

$$[\![\mathsf{P} \xrightarrow{k} \mathsf{P}' \big| X]\!] := \max_{\substack{\mathbf{x} \in X^k \\ \hat{\mathbf{y}}, D}} \|\mathsf{P}'|_{D|^{\mathbf{x}}}\, \mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}}\, \mathsf{P}|_{D|^{\mathbf{x}}}\|. \tag{12}$$

where the max is restricted to $\mathbf{x} \in X^k$. Obviously, $[\![\mathsf{P} \xrightarrow{k} \mathsf{P}']\!] = [\![\mathsf{P} \xrightarrow{k} \mathsf{P}' \big| \mathcal{X}]\!]$.

Lemma 5.26. *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*

$$[\![\mathsf{P} \xrightarrow{k} \mathsf{P}'' \big| X]\!] \leq [\![\mathsf{P} \xrightarrow{k} \mathsf{P}'' \backslash \mathsf{Q} \big| X]\!] + [\![\mathsf{P} \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}'' \big| X]\!], \tag{13}$$

where furthermore

$$[\![\mathsf{P} \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}'' \big| X]\!] \leq [\![\mathsf{P} \xrightarrow{k'} \neg\mathsf{Q} \big| X]\!] + [\![\mathsf{P} \xrightarrow{k'} \mathsf{Q} \cap \mathsf{P}' \big| X]\!] + [\![\mathsf{Q} \backslash \mathsf{P}' \xrightarrow{k''} \mathsf{Q} \cap \mathsf{P}'' \big| X]\!] \tag{14}$$

as well as

$$[\![\mathsf{P} \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}'' \big| X]\!] \leq [\![\mathsf{P} \xrightarrow{k} \neg\mathsf{Q} \big| X']\!] + [\![\mathsf{P} \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}' \big| X']\!] + [\![\mathsf{Q} \backslash \mathsf{P}' \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}'' \big| X'']\!]. \tag{15}$$

Proof. The first inequality follows immediately from Lemma 5.24, using that $(\mathsf{P}'' \setminus \mathsf{Q}) \cup (\mathsf{Q} \cap \mathsf{P}'') = \mathsf{P}''$. For the other two, let $\mathbf{x} \in X^k$, $\hat{\mathbf{y}} \in \hat{\mathcal{Y}}^k$, $D \in \mathfrak{D}$ be the choices that achieve the maximal value in the definition of $[\![\mathsf{P} \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}'' \big| X]\!]$. We may assume without loss of generality that $\mathbf{x}$ consists of pairwise distinct entries. For proving the first inequality, we split up $\mathbf{x}$ into $(\mathbf{x}', \mathbf{x}'') \in X^{k'} \times X^{k''}$, and correspondingly then for $\hat{\mathbf{y}} \in \hat{\mathcal{Y}}^k$. For proving the second inequality, we let $\mathbf{x}'$ consist of all coordinates of $\mathbf{x}$ that lie in X', and we let $\mathbf{x}''$ consist of all coordinates of $\mathbf{x}$ that lie in X'' but not in X', and $\hat{\mathbf{y}}'$ and $\hat{\mathbf{y}}''$ consists of the corresponding coordinates of $\hat{\mathbf{y}}$; in this case, $(\mathbf{x}', \mathbf{x}'') \in X'^{\ell'} \times X''^{\ell''}$ with $\ell_0 + \ell_1 = k$. In both cases, we have $\mathsf{cO}_{\mathbf{x}\hat{\mathbf{y}}} = \mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\mathsf{cO}_{\mathbf{x}''\hat{\mathbf{y}}''}$, and, writing $\mathsf{P}_{\mathbf{x}}$ for $\mathsf{P}|_{D|^{\mathbf{x}}}$ etc., we obtain

$$\begin{aligned}
&[\![\mathsf{P} \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}'' \big| X]\!] = \|\mathsf{P}''_{\mathbf{x}}\mathsf{Q}^{\mathbf{x}}\, \mathsf{cO}_{\mathbf{x}''\hat{\mathbf{y}}''}\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\, \mathsf{P}_{\mathbf{x}}\| \\
&\quad \leq \|\mathsf{P}''_{\mathbf{x}}\mathsf{Q}^{\mathbf{x}}\, \mathsf{cO}_{\mathbf{x}''\hat{\mathbf{y}}''}\mathsf{Q}^{\mathbf{x}}\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\, \mathsf{P}_{\mathbf{x}}\| + \|(\mathsf{I} - \mathsf{Q}^{\mathbf{x}})\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\, \mathsf{P}_{\mathbf{x}}\| \\
&\quad \leq \|\mathsf{P}'_{\mathbf{x}}\mathsf{Q}^{\mathbf{x}}\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\, \mathsf{P}_{\mathbf{x}}\| + \|\mathsf{P}''_{\mathbf{x}}\mathsf{Q}^{\mathbf{x}}\, \mathsf{cO}_{\mathbf{x}''\hat{\mathbf{y}}''}(\mathsf{I} - \mathsf{P}'_{\mathbf{x}})\mathsf{Q}^{\mathbf{x}}\| + \|(\mathsf{I} - \mathsf{Q}^{\mathbf{x}})\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\, \mathsf{P}_{\mathbf{x}}\| \\
&\quad \leq \|\mathsf{P}'_{\mathbf{x}'}\mathsf{Q}^{\mathbf{x}'}\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\, \mathsf{P}_{\mathbf{x}'}\| + \|\mathsf{P}''_{\mathbf{x}''}\mathsf{Q}^{\mathbf{x}''}\, \mathsf{cO}_{\mathbf{x}''\hat{\mathbf{y}}''}(\mathsf{I} - \mathsf{P}'_{\mathbf{x}''})\mathsf{Q}^{\mathbf{x}''}\| + \|(\mathsf{I} - \mathsf{Q}^{\mathbf{x}'})\mathsf{cO}_{\mathbf{x}'\hat{\mathbf{y}}'}\, \mathsf{P}_{\mathbf{x}'}\|,
\end{aligned}$$

where the last equality follows from basic properties of the operator norm. The first of the two remaining bounds is now obtained by maximizing the individual terms on the right hand side over $\mathbf{x}' \in X^{k'}$ and $\mathbf{x}'' \in X^{k''}$ (as well as over $\hat{\mathbf{y}}', \hat{\mathbf{y}}''$ and D). For the other case, we maximize over $\mathbf{x}' \in X'^{\ell'}$ and $\mathbf{x}'' \in X''^{\ell''}$ and exploit that, for instance, $[\![\mathsf{P} \xrightarrow{\ell'} \neg\mathsf{Q} \big| X']\!] \leq [\![\mathsf{P} \xrightarrow{k} \neg\mathsf{Q} \big| X']\!]$, given that $\ell' \leq k$. □

By recursive application of Lemma 5.26, we obtain the following.

Corollary 5.27 (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*

$$\left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \mathsf{P}_h \middle| X\right]\!\right] \leq \sum_{i=1}^{h} \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{\bar{k}_i} \neg\mathsf{Q} \middle| X\right]\!\right] + \sum_{i=1}^{h} \left[\!\left[\mathsf{Q}\backslash\mathsf{P}_{i-1} \xrightarrow{k_i} \mathsf{Q} \cap \mathsf{P}_i \middle| X\right]\!\right] \qquad \textit{and}$$

$$\left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \mathsf{P}_h \middle| X\right]\!\right] \leq \sum_{i=1}^{h} \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \neg\mathsf{Q} \middle| \bar{X}_i\right]\!\right] + \sum_{i=1}^{h} \left[\!\left[\mathsf{Q}\backslash\mathsf{P}_{i-1} \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}_i \middle| X_i\right]\!\right],$$

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

Proof. Applying (13) and (14) with $\mathsf{P} := \neg\mathsf{P}_0$, $\mathsf{P}' := \mathsf{P}_{h-1}$ and $\mathsf{P}'' := \mathsf{P}_h$, and omitting the "conditioning" on X for simplicity, we get

$$\left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \mathsf{P}_h\right]\!\right] \leq \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \neg\mathsf{Q}\right]\!\right] + \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{\bar{k}_{h-1}} \neg\mathsf{Q}\right]\!\right] + \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{\bar{k}_{h-1}} \mathsf{Q} \cap \mathsf{P}_{h-1}\right]\!\right] + \left[\!\left[\mathsf{Q}\backslash\mathsf{P}_{h-1} \xrightarrow{k_h} \mathsf{Q} \cap \mathsf{P}_h\right]\!\right].$$

Recursively applying (14) to $\left[\!\left[\neg\mathsf{P}_0 \xrightarrow{\bar{k}_{h-1}} \mathsf{Q} \cap \mathsf{P}_{h-1}\right]\!\right]$ gives the first claim. The second is argued correspondingly. □

The quantum transition capacity *with restricted input*, defined in (12), is just the original definition of the quantum transition capacity (Definition 5.5) but with the considered set $\mathcal{X}$ replaced by X. As a consequence, properties for $\left[\!\left[\mathsf{P} \to \mathsf{P}'\right]\!\right]$ carry over to $\left[\!\left[\mathsf{P} \to \mathsf{P}' \middle| X\right]\!\right]$. For instance, it is still symmetric, and Lemma 5.24 carries over to

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

etc. For completeness we spell out here the definition of non-uniform recognizability as well as Theorem 5.19 for such input-restricted database transitions $\mathsf{P} \to \mathsf{P}' \mid X$ (the other types of recognizability can be generalized similarly).

Definition 5.28. *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*

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

Theorem 5.29. *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*

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

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 [8] (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 [9]. 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 Appendix B. Typically, underlying the construction of a PoSW is a directed acyclic graph (DAG) G with certain "depth-robust" properties, and a graph labelling 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 labelling.

Simple PoSW DAG and Graph Labelling. 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 6.2), 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 6.1. *We write* $\mathsf{rt} := \epsilon$ *for the* root*, and we write* $\mathsf{leaves}(V_n) := \{0,1\}^n$ *for the* leaves *in* V_n*. For* $T \subseteq V_n$*, we set* $\mathsf{leaves}(T) := T \cap \{0,1\}^n$*. For* $v \notin \mathsf{leaves}(V_n)$*, let* $\mathsf{left}(v) := 0\|v$ *and* $\mathsf{right}(v) := 1\|v$*. For* $b \in \{0,1\}$ *and* $v \in \{0,1\}^{<n}$*, let* $\mathsf{par}(b\|v) := v$ *and* $\mathsf{sib}(b\|v) := \neg b\|v$ *(see Fig. 2, right). Finally, for a leaf* $v \in \mathsf{leaves}(V_n)$*, we define the* authentication path *of* v *(as in the Merkle tree) as* $\mathsf{ap}(v) = \{\mathsf{par}^i(v), \mathsf{sib}(\mathsf{par}^i(v)) : 0 \leq i < n\}$.

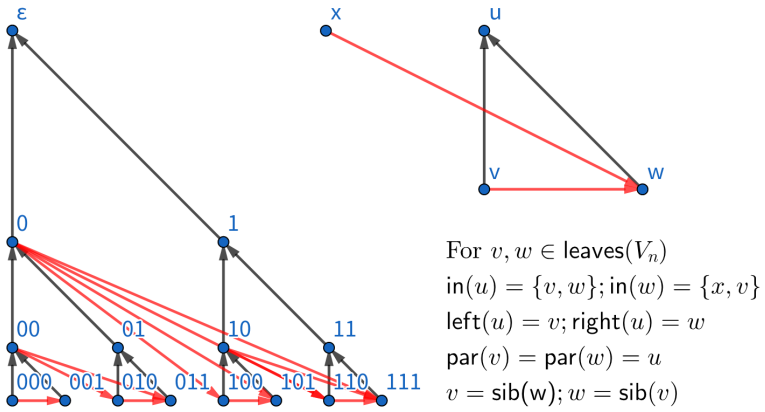

Figure 2: Illustration of the Simple PoSW DAG G_n^{PoSW} for $n = 3$.

Definition 6.2. *For* $n \in \mathbb{N}$*, define the Simple PoSW DAG* $G_n^{\mathsf{PoSW}} := (V_n, E'_n \cup E''_n)$ *with vertex set* V_n *and edges*

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

For $v \in V_n$, we write $\mathsf{in}(v) := \{u \in V_n \,|\, (u,v) \in E'_n \cup E''_n\}$ 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 labelling 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 being well defined).

Definition 6.3 (Graph Labelling)**.** *A function* $\ell : V_n \to \{0,1\}^w$*,* $v \mapsto \ell_v$ *is a called a* labelling *of* G_n^{PoSW} *with respect to* H *if*

$$\ell_v = H(v, \ell_{\mathsf{in}(v)}) \tag{16}$$

for all $v \in V_n$*, were* $\ell_{\mathsf{in}(v)}$ *is shorthand for* $(\ell_{v_1}, \ldots, \ell_{v_d})$ *with* $\{v_1, \ldots, v_d\} = \mathsf{in}(v)$*. Similarly, for a subtree*[17] T *of* G_n^{PoSW}*, a function* $\ell : T \to \{0,1\}^w$*,* $v \mapsto \ell_v$ *is a called a* labelling *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 the structure of the graph, G_n^{PoSW} admits a unique labelling, which can be computed by making $N = 2^{n+1} - 1$ sequential queries to H, starting with the leftmost leaf. We sometimes speak of a *consistent* labelling (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 (16) then in particular means that $H(v, \ell_{\mathsf{in}(v)}) \neq \bot$.

We also make the following important remark.

[17] 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.

Remark 6.4. Let T be a subtree of G_n^{PoSW} with a consistent labelling ℓ. Then, any path $P = (v_0, \ldots, v_r)$ of length $|P| = r$ in T induces an r-chain $(x_0, \ldots, x_r)$, where $x_i = (v_i, \ell_{v'_1}, \ldots, \ell_{v'_d})$ with $\{v'_1, \ldots, v'_d\} = \mathsf{in}(v_i)$, and where the relation $\triangleleft$ is defined as follows. $y \triangleleft x$ if and only if x is of form $(v, \ell_1, \ell_2, \ldots, \ell_d)$ 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. We are ready to describe the (non-interactive) Simple PoSW construction, which amounts to asking the prover $\mathcal{P}$ to compute the root label of G_n^{PoSW} with respect to the hash function 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 the challenge 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 labelling 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: $\gamma := H_\chi^{\mathsf{ChQ}}(\phi) := \big(H_\chi(\phi, 1), \ldots, H_\chi(\phi, d)\big) \in \{0,1\}^{dw}$ for sufficiently large d, parsed as t leaves $\{v_1, \ldots, v_t\} \subseteq \mathsf{leaves}(V_n)$.
- $\tau := \mathsf{open}^H(\chi, N, \phi_{\mathcal{P}}, \gamma)$: For challenge $\gamma = \{v_1, \ldots, v_t\}$, the opening τ consists of the labels of 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 the consistency of the labelled authentication paths $\mathsf{ap}(v_i)$. Specifically, for each $i \in [t]$ and $0 \leq j \leq n$, $\mathcal{V}$ checks if $\ell_u = H_\chi(u, \ell_{\mathsf{in}(u)})$ for $u = \mathsf{par}^j(v_i)$. $\mathcal{V}$ 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 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. Before presenting our proof of post-quantum security for Simple PoSW, we first review the classical security analysis in [8]. For simplicity, here we consider the original (interactive) Simple PoSW (i.e., $\mathcal{P}$ first sends ϕ, receives random γ from $\mathcal{V}$, and then sends τ to $\mathcal{V}$). Also, for now, we assume that $\mathcal{P}$ does not make further oracle queries after sending ϕ. We review the argument of [8] 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 $D : \{0,1\}^{\leq B} \to \{0,1\}^w \cup \{\bot\}$ be the database at the point that $\mathcal{A}$ sends ϕ to $\mathcal{V}$ (after making the q k-parallel queries). Following the argument in Section 2, we can bound the success probability of $\mathcal{A}$ by bounding the probability that a random challenge $\gamma = \{v_i\}_{i \in [t]}$ can be opened based on the information in the database D. As argued in Section 2, the probability that the database D contains collisions, or a $(q+1)$-chain with respect to the relation defined in Remark 6.4, is small; specifically, at most $O((k^2q^2 + nkq^2)/2^w)$. Thus, by a union bound, 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, i.e., for which he can provide a consistently labelled authentication path $\mathsf{ap}(v)$. One of the key observations in [8] 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 labelling ℓ with $\ell_{\mathsf{rt}} = \phi$. As observed in [8], this subtree T then contains all leaves that one can open given D. Thus, $\mathcal{A}$ can correctly answer a challenge $\gamma = \{v_1, \ldots, v_t\}$ if and only if $\gamma \subseteq \mathsf{leaves}(T)$.

The subtree T, together with the labelling ℓ of T, can be extracted using $\mathsf{Extract}_n^D(\phi)$, described in Algorithm 1 in the Appendix C. 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 $v \in T$ with $\mathsf{in}(v) \subseteq T$ check if $\ell_v = D(v, \ell_{\mathsf{in}(v)})$ and remove v from T if this is not the case.

The last step is to bound the number of leaves in T. Another key argument in [8] 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 $|P| \geq 2 \cdot |\mathsf{leaves}(T)| - 2$. Recall we argued above that the graph labelling of a path $P \in G_n^{\mathsf{PoSW}}$ induces a $|P|$-chain in H. The same argument applies here to show that there exists a $|P|$-chain in D since the extracted labels in $P \subseteq T$ are consistent (i.e., satisfying $D(v, \ell_{\mathsf{in}(v)}) = \ell_v$). Combining these with the assumption that D contains no $q+1$-chain, we have $|\mathsf{leaves}(T)| \leq (q+2)/2$. Therefore, the probability that $\mathcal{A}$ can open labels for a random challenge $\gamma = \{v_i\}_{i \in [t]}$ is at most

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

Finally, we briefly discuss here how to handle the case that $\mathcal{A}$ can make additional queries after sending ϕ, as a similar argument is required in the analysis of the non-interactive Simple PoSW in the next section. As before, let D be the database right after $\mathcal{A}$ has sent $\phi = \ell_{\mathsf{rt}}$, but now $\mathcal{A}$ can make additional queries after seeing γ, which adds new entries to D and may help $\mathcal{A}$ to open labels for more challenges γ.

The main observation to analyze whether additional queries are helpful is as follows. Recall that T contains all leaves v that admit a consistently labelled authentication path $\mathsf{ap}(v)$. 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 $D(x) = \perp$ 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 $O(qk)$.

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 6.5 (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 that $\tilde{\mathcal{P}}$ can make the verifier $\mathcal{V}$ accept is at most*

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

The first step towards the proof is to invoke Theorem 5.6, which, in the case here, bounds the success probability p of a dishonest prover $\tilde{\mathcal{P}}$ by

$$\sqrt{p} \leq \left[\!\left[\perp \xRightarrow{k,q} \mathsf{P}^R \right]\!\right] + \sqrt{\frac{t \cdot (n+1) + 1}{2^w}} ,$$

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 $\triangleleft$ considered in Remark 6.4. By the properties of (the subtree extracted with) $\mathsf{Extract}_n^D(\cdot)$, we have

$$\mathsf{Suc} \setminus \mathsf{CL} = \left\{ D \in \neg\mathsf{CL} \,\middle|\, \exists\, \ell_{\mathsf{rt}} \in \{0,1\}^w \text{ s.t. } D^{\mathsf{ChQ}}(\ell_{\mathsf{rt}}) \subseteq \mathsf{Extract}_n^D(\ell_{\mathsf{rt}}) \right\}. \tag{17}$$

To bound the above quantum transition capacity $\left[\!\left[\perp \xRightarrow{k,q} \mathsf{P}^R \right]\!\right] = \left[\!\left[\perp \xRightarrow{k,q} \mathsf{Suc} \right]\!\right]$, we consider database properties $\mathsf{P}_0, \ldots, \mathsf{P}_q$ with $\mathsf{P}_0 = \perp$ and $\mathsf{P}_s = \mathsf{Suc} \cup \mathsf{CL} \cup \mathsf{CHN}^{s+1}$ for $1 \leq s \leq q$. Following Remark 5.7 and using Corollary 5.25,

$$\left[\!\left[\perp \xRightarrow{k,q} \mathsf{Suc} \right]\!\right] \leq \left[\!\left[\perp \xRightarrow{k,q} \mathsf{P}_q \right]\!\right] \leq \sum_{1 \leq s \leq q} \left[\!\left[\mathsf{SZ}_{\leq k(s-1)} \setminus \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s \right]\!\right].$$

Thus, the proof of Theorem 6.5 follows immediately from the following bound on the considered transition capacity.

Proposition 6.6. *For integers $0 \le s \le q$, and for the database properties $\mathsf{P}_0, \ldots, \mathsf{P}_q$ as defined above*

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

Intuitively, we consider the transition from a database that is bounded in size, has no collision, no s-chain and does not have a successful output for $\tilde{\mathcal{P}}$, into one that contains a collision or an $(s+1)$-chain or a successful output for $\tilde{\mathcal{P}}$.

Proof. Applying Corollary 5.27 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 in Corollary 5.27 set to[18]

$$\neg\mathsf{P}_0 := \mathsf{SZ}_{\le k(s-1)} \setminus \mathsf{P}_{s-1}\,, \quad \mathsf{P}_1 := \mathsf{Suc}\,, \quad \mathsf{P}_2 := \mathsf{Suc} \cup \mathsf{CL} \cup \mathsf{CHN}^{s+1} = P_s \quad \text{and} \quad \mathsf{Q} := \neg(\mathsf{CL} \cup \mathsf{CHN}^{s+1})\,,$$

we can bound $\left[\!\left[\mathsf{SZ}_{\le k(s-1)} \backslash \mathsf{P}_{s-1} \xrightarrow{k} \mathsf{P}_s\right]\!\right] = \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \mathsf{P}_2\right]\!\right]$ by

$$\begin{aligned}
&\le \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \neg\mathsf{Q} \middle| \mathsf{LbQ}\right]\!\right] + \left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \neg\mathsf{Q} \middle| \mathsf{ChQ} \cup \mathsf{LbQ}\right]\!\right] + \left[\!\left[\mathsf{Q}\backslash\mathsf{P}_0 \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}_1 \middle| \mathsf{LbQ}\right]\!\right] + \left[\!\left[\mathsf{Q}\backslash\mathsf{P}_1 \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}_2 \middle| \mathsf{ChQ}\right]\!\right] \\
&\le 2\left[\!\left[\neg\mathsf{P}_0 \xrightarrow{k} \neg\mathsf{Q}\right]\!\right] + \left[\!\left[\mathsf{Q}\backslash\mathsf{P}_0 \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}_1 \middle| \mathsf{LbQ}\right]\!\right] + \left[\!\left[\mathsf{Q}\backslash\mathsf{P}_1 \xrightarrow{k} \mathsf{Q} \cap \mathsf{P}_2 \middle| \mathsf{ChQ}\right]\!\right] \\
&= 2\left[\!\left[\mathsf{SZ}_{\le k(s-1)}\backslash\mathsf{P}_{s-1} \xrightarrow{k} \mathsf{CL} \cup \mathsf{CHN}^{s+1}\right]\!\right] + \left[\!\left[\mathsf{SZ}_{\le k(s-1)}\backslash\mathsf{P}_{s-1}\backslash\mathsf{CL}\backslash\mathsf{CHN}^{s+1} \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL}\backslash\mathsf{CHN}^{s+1} \middle| \mathsf{LbQ}\right]\!\right] \\
&\qquad + \left[\!\left[\neg(\mathsf{Suc} \cup \mathsf{CL} \cup \mathsf{CHN}^{s+1}) \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL}\backslash\mathsf{CHN}^{s+1} \middle| \mathsf{ChQ}\right]\!\right] \\
&\le 2\left[\!\left[\mathsf{SZ}_{\le k(s-1)}\backslash\mathsf{P}_{s-1} \xrightarrow{k} \mathsf{CL} \cup \mathsf{CHN}^{s+1}\right]\!\right] + \left[\!\left[\mathsf{SZ}_{\le k(s-1)}\backslash\mathsf{P}_{s-1} \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL} \middle| \mathsf{LbQ}\right]\!\right] + \left[\!\left[\neg\mathsf{P}_s \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL} \middle| \mathsf{ChQ}\right]\!\right].
\end{aligned}$$

By means of Lemma 5.24 (and Corollary 5.25), 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

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

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

Intuitively, $\left[\!\left[\neg\mathsf{P}_s \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL} \middle| \mathsf{ChQ}\right]\!\right]$ captures the likelihood that a database $D \notin \mathsf{Suc}$ (and with no collision and chain) is tuned into one that does satisfy Suc by (re)defining D on k values that correspond to challenge queries. For this to happen, one of the newly defined function values of D, corresponding to a challenge query and thus specifying a set of leaves, must "hit" the set of leaves that can be answered, which is bounded in size.

Lemma 6.7. *For any positive integer q, it holds that $\left[\!\left[\neg\mathsf{P}_q \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL} \middle| \mathsf{ChQ}\right]\!\right] \le ek \cdot \sqrt{10\left(\frac{q+2}{2^{n+1}}\right)^t}$.*

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

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

Suppose $D_{\mathbf{x},\mathbf{r}} \in \neg\mathsf{P}_q = \mathsf{Fail}\backslash\mathsf{CL} \setminus \mathsf{CHN}^{q+1}$ but $D_{\mathbf{x},\mathbf{u}} \in \mathsf{Suc} \setminus \mathsf{CL}$. Thus, by (17), there exists $\ell_{\mathsf{rt}} \in \{0,1\}^w$ with

$$D_{\mathbf{x},\mathbf{u}}^{\mathsf{ChQ}}(\ell_{\mathsf{rt}}) \subseteq \mathsf{leaves}\Big(\mathsf{Extract}_n^{D_{\mathbf{x},\mathbf{u}}}(\ell_{\mathsf{rt}})\Big)\,, \tag{18}$$

[18]Note that we have slight collision of notation here: $\mathsf{P}_0, \mathsf{P}_1, \mathsf{P}_2$ correspond to the choice of properties for applying Corollary 5.27, and should not be confused with P_s with s set to $0, 1, 2$, respectively.

while

$$D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{r}}(\ell_{\mathsf{rt}}) \not\subseteq \mathsf{leaves}\Big(\mathsf{Extract}^{D_{\mathbf{x},\mathbf{r}}}_n(\ell_{\mathsf{rt}})\Big). \tag{19}$$

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

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

If ℓ_{rt} is different from all ℓ^j_{rt}, then equations (18) and (19) contradict. So there is some j such that $\ell^j_{\mathsf{rt}} = \ell_{\mathsf{rt}}$. Equations (18) and (19) thus become

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

understanding that taking u_j and r_j represent lists/sets of t (challenge) leaves. Hence $r_j \neq u_j$. This concludes that $\mathsf{L}^{\mathbf{x},D}_j$ 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, recycling an element from the classical reasoning, we have

$$\Big|\mathsf{leaves}\Big(\mathsf{Extract}^{D_{\mathbf{x},\perp}}_n(\ell^j_{\mathsf{rt}})\Big)\Big| \leq \frac{q+2}{2}.$$

Therefore,

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

and so the claimed bound follows by applying Theorem 5.29. □

Similarly here, the intuition is that $\big[\!\big[\neg\mathsf{P}_s \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL}\big|\mathsf{LbQ}\big]\!\big]$ captures the likelihood that a database $D \notin \mathsf{Suc}$ (and with no collision and chain) is tuned into one that does satisfy Suc by (re)defining D on k values that correspond to label queries. For this to happen, one of the newly defined function values of D, corresponding to a label, must "match up" with the other labels.

Lemma 6.8. *For any positive integer q, it holds that $\big[\!\big[\mathsf{SZ}_{\leq k(q-1)}\backslash\mathsf{P}_{q-1} \xrightarrow{k} \mathsf{Suc}\backslash\mathsf{CL}\big|\mathsf{LbQ}\big]\!\big] \leq ek\sqrt{\frac{10nkq}{2^w}}$.*

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

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

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

In order to bound above capacity, we define the 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

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

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 (17), there exists ℓ_{rt} so that $D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{u}}(\ell_{\mathsf{rt}}) \subseteq \mathsf{Extract}^{D_{\mathbf{x},\mathbf{u}}}_n(\ell_{\mathsf{rt}})$, while, on the other hand, there exists some $v \in D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{r}}(\ell_{\mathsf{rt}}) \setminus \mathsf{leaves}\big(\mathsf{Extract}^{D_{\mathbf{x},\mathbf{r}}}_n(\ell_{\mathsf{rt}})\big)$. 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 (18), we have

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

Consider the partial labelling ℓ and the subgraph T extracted from $\mathsf{Extract}^{D_{\mathbf{x},\mathbf{u}}}_n(\ell_{\mathsf{rt}})$. Then, given that $v \in \mathsf{leaves}(T)$, the labeling ℓ is in particular a consistent labeling of the authentication path $\mathsf{ap}(v)$ with respect to $D_{\mathbf{x},\mathbf{u}}$, i.e., $\ell_{z_i} =$

$D_{\mathbf{x},\mathbf{u}}(z_i, \ell_{\mathsf{in}(z_i)})$ for $z_i = \mathsf{par}^i(v)$ and $0 \leq i \leq n$. Furthermore, $D^{\mathsf{ChQ}}_{\mathbf{x},\mathbf{u}}(\ell_{\mathsf{rt}}) \neq \bot$. Therefore, $\ell_{z_i} \in \mathrm{LSupp}(D_{\mathbf{x},\mathbf{u}}) \subseteq \mathrm{LSupp}(D_{\mathbf{x},\mathbf{0}})$ for $0 \leq i \leq n$.

On another hand, since v is not a leaf extracted from $D_{\mathbf{x},\mathbf{r}}$, yet $D_{\mathbf{x},\mathbf{r}}(\ell_{\mathsf{rt}}) = D_{\mathbf{x},\mathbf{u}}(\ell_{\mathsf{rt}})$, it must be that ℓ is *not* a consistent labelling of $\mathsf{ap}(v)$ with respect to $D_{\mathbf{x},\mathbf{r}}$. Therefore, there must exist i and j such that $x_j = (z_i, \ell_{\mathsf{in}(z_i)})$ and $u_j = D_{\mathbf{x},\mathbf{u}}(x_j) = \ell_{z_i} \neq D_{\mathbf{x},\mathbf{r}}(x_j) = r_j$. Thus, $r_j \neq u_j$ and $u_j \in \mathrm{LSupp}(D_{\mathbf{x},\mathbf{0}})$. Therefore $\mathsf{L}_j^{\mathbf{x},D}$ 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

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

and thus the claimed bound follows from applying Theorem 5.29. □

References

[1] Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1(1):37–46, 2005.

[2] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[3] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[4] Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. 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] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Lee D.H. and Wang X., editors, *Advances in Cryptology ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

[6] Gilles Brassard, Peter Hoyer, and Alain Tapp. Quantum algorithm for the collision problem. *arXiv preprint quant-ph/9705002*, 1997.

[7] Canetti, Goldreich, and Halevi. The random oracle methodology, revisited (preliminary version). In *ACM Symposium on Theory of Computing (STOC)*, 1998.

[8] Bram Cohen and Krzysztof Pietrzak. Simple proofs of sequential work. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 451–467. Springer, 2018.

[9] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986.

[10] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[11] Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019*, volume 11693 of *Lecture Notes in Computer Science*, pages 326–355. Springer, 2019.

[12] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 129–146. Springer, 2014.

[13] Christof Zalka. Grover's quantum searching algorithm is optimal. *Phys. Rev. A*, 60:2746–2751, Oct 1999.

[14] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268. Springer, 2019.

A Efficient Simulation of the Compressed Oracle

In order to complete our exposition of the compressed oracle, we show here another aspect of the technique, which is not relevant in our context but an important feature in other applications: similarly to the classical lazy-sampling technique, the evolution of the compressed oracle can be *efficiently* computed, *and* useful information can be *efficiently* extracted from the compressed oracle.

For concreteness, we assume here that $\mathcal{Y} = \{0,1\}^m$. This in particular means that $\hat{\mathcal{Y}} = \mathcal{Y}$, and that there is a designated and efficiently computable *quantum Fourier transform* $\mathsf{QFT} : |y\rangle \mapsto |\hat{y}\rangle = H^{\otimes m}|y\rangle$. This then also means that $\mathfrak{D} = \hat{\mathfrak{D}}$, but we still distinguish between $|D\rangle = \bigotimes_x |D(x)\rangle$ and $|\hat{D}\rangle = \bigotimes_x \mathsf{QFT}|D(x)\rangle$ for any $D \in \mathfrak{D}$. Additionally, we assume that $\mathcal{X}$ comes with an efficiently computable total order, say $\mathcal{X} = \{0,1\}^n$.

Consider the classical encoding function $Enc : \mathfrak{D} \to \mathfrak{L} := ((\mathcal{X} \times \mathcal{Y}) \cup \{\bot\})^{|\mathcal{X}|}$ that maps $D \in \mathfrak{D}$ to the list $L = \big[(x_1, y_1), \ldots, (x_s, y_s), \bot, \ldots, \bot\big]$ of pairs (x_i, y_i) for which $y_i = D(x_i) \neq \bot$, sorted as $x_1 < \cdots < x_s$ and padded with $\bot$'s. Recall the unitary cO, defined in Section 4.3 and which describes the evolution of the compressed oracle, and consider the corresponding "update function" $Upd : \mathcal{X} \times \mathcal{Y} \times \mathfrak{L} \to \mathcal{X} \times \mathcal{Y} \times \mathfrak{L}$, defined to satisfy

$$\begin{aligned} Upd\big(x, y, Enc(D)\big) &= \big(x, y, Enc(D')\big) \\ &\iff |x\rangle|\hat{y}\rangle|\hat{D}'\rangle = \mathsf{cO}|x\rangle|\hat{y}\rangle|\hat{D}\rangle = |x\rangle|\hat{y}\rangle \otimes \mathsf{cO}_{x\hat{y}}|\hat{D}\rangle \end{aligned}$$

for any $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $D \in \mathfrak{D}$. By construction, and exploiting (7), it turns out that Upd is a rather simple function. Applied to $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and $L = [(x_1, y_1), \ldots, (x_s, y_s), \bot, \ldots, \bot] \in \mathfrak{L}$, it acts as follows. If $y_i = 0$ for some i then it acts as identity,[19] otherwise, the following two cases are distinguished: if $x \notin \{x_1, \ldots, x_s\}$ and $y \neq 0$ then Upd inserts the pair (x, y) to the list, while if $x = x_i$ and $y \neq y_i$ for some i then Upd replaces (x_i, y_i) by $(x_i, y_i \oplus y)$. In particular, for lists L of bounded size $s \leq Q$, the classical function Upd can be efficiently computed, i.e., in time polynomial in Q and in the size of the bit representations of the elements of $\mathcal{X}$ and $\mathcal{Y}$.

Formally, for a fixed Q, let $\mathfrak{D}_{\leq Q} := \{D \in \mathfrak{D} \,:\, |\{x \in \mathcal{X} \,:\, D(x) = \bot\}| \leq Q\}$, and let $enc : \mathfrak{D}_{\leq Q} \to \mathfrak{L}_{\leq Q} := ((\mathcal{X} \times \mathcal{Y}) \cup \{\bot\})^Q$ be defined in the obvious way, i.e., so that $enc(D)$ is obtained from $Enc(D)$ by removing the rightmost $\bot$-paddings. Similarly, $upd : \mathcal{X} \times \mathcal{Y} \times \mathfrak{L}_{\leq Q} \to \mathcal{X} \times \mathcal{Y} \times \mathfrak{L}_{\leq Q}$ is defined in the obvious way to coincide with Upd except for the shorter $\bot$-padding, and *except for the following additional modification*: upd is declared to act as identity on (x, y, L) whenever $s = Q$ and $x \neq \{x_1, \ldots, x_s\}$, i.e., when there would be an "overflow". It then follows that upd is an *efficiently computable permutation*. Thus, by basic theory of quantum computation, the corresponding unitary $|x, y, L\rangle \mapsto |upd(x, y, L)\rangle$ can be efficiently computed by means of a polynomial sized quantum circuit. Hence, by means of the encoding

$$\hat{\mathsf{enc}} : |\hat{D}\rangle \mapsto |enc(D)\rangle = |x_1\rangle|y_1\rangle \cdots |x_s\rangle|y_s\rangle|\bot\rangle \cdots |\bot\rangle\,,$$

the unitary cO can be efficiently computed, as long as it acts on $\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{Y}] \otimes \mathbb{C}[\mathfrak{D}_{<Q}]$, i.e., as long as fewer than Q queries are being made.

Alternatively, we can also consider the following variant, where $|D\rangle$, rather than $|\hat{D}\rangle$, is encoded as $|enc(D)\rangle$:

$$\mathsf{enc} : |D\rangle \mapsto |enc(D)\rangle = |x_1\rangle|y_1\rangle \cdots |x_s\rangle|y_s\rangle|\bot\rangle \cdots |\bot\rangle$$

This encoding offer the following useful property. Consider a unitary U_f on $\mathbb{C}[\mathfrak{D}]$, plus an ancilla, that computes a classical function f, meaning that $U_f : |D\rangle|w\rangle \mapsto |D\rangle|w \oplus f(D)\rangle$, and for which the classical function f is efficiently computable for $D \in \mathfrak{D}_{\leq Q}$ and given that D is represented by $enc(D)$. Then, the unitary U_f is efficiently computable with the considered encoding enc. This allows for efficient extraction of useful information from the compressed oracle. Typical examples would be to check whether a certain preimage $x_\circ \in \mathcal{X}$ is in the database, i.e., whether $D(x_\circ) \neq \bot$, or to check whether there is a 0-preimage in the database, i.e. whether $\exists\, x : D(x) = 0$, etc.

The final, simple yet crucial, observation is that one can efficiently switch between these two encodings. Indeed, it is easy to see that, say, enc commutes with applying the quantum Fourier transform QFT in the obvious way, i.e.,

$$\mathsf{enc}|\hat{D}\rangle = |x_1\rangle|\hat{y}_1\rangle \cdots |x_s\rangle|\hat{y}_s\rangle|\bot\rangle \cdots |\bot\rangle\,.$$

[19] This is the "artificial" case, which was introduced to have cO defined on the entire space $\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{Y}] \otimes \mathbb{C}[\mathfrak{D}]$

Thus, enc equals ênc up to some QFT's to be applied (controlled by the corresponding register not being $\perp$), which can be efficiently done. Hence, by a suitable encoding, both the evolution of the compressed oracle as well as efficiently computable classical functions on the (suitably encoded) database D, can be efficiently computed by a quantum circuit.

B PoSW Definition

Informally, a (non-interactive) PoSW allows a prover $\mathcal{P}$ to generate an efficiently verifiable proof showing that some computation was going on for N sequential steps since some "statement" χ was received, in the sense that even a powerful adversary with parallel computation power cannot compute a valid proof with much less than N steps. PoSW is typically constructed in the random oracle model. We recall its formal definition from [8] (after applying the Fiat-Shamir transformation) as follows (see Figure 3 for an illustration).

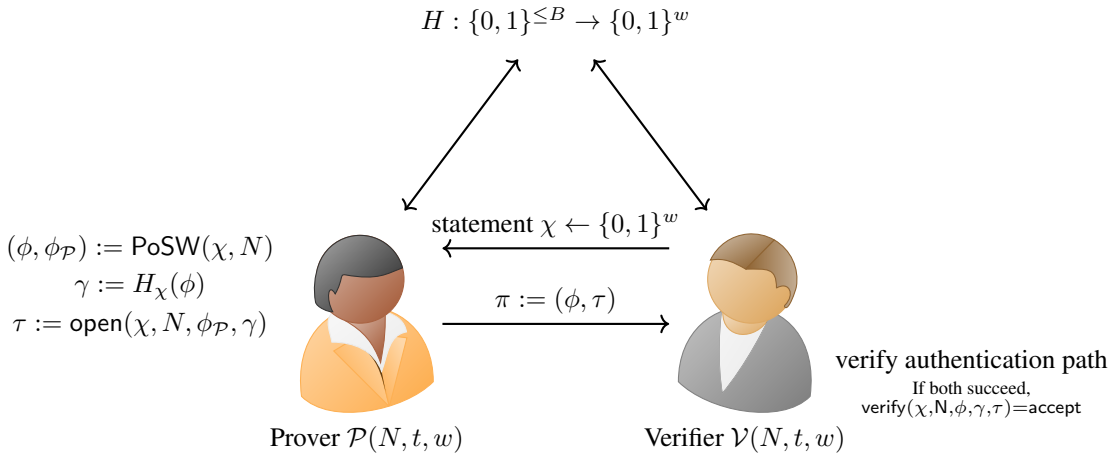

Figure 3: Non-interactive PoSW.

- *Common Inputs:* The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ get as common input two statistical security parameters $w, t \in \mathbb{N}$ and a time parameter $N \in \mathbb{N}$. They have access to a random oracle $H : \{0,1\}^{\leq B} \to \{0,1\}^w$, where B is sufficiently large but otherwise arbitrary.[20]
- *Statement:* $\mathcal{V}$ samples a random $\chi \leftarrow \{0,1\}^w$ and sends it to $\mathcal{P}$.
- *Compute PoSW:* $\mathcal{P}$ computes $(\phi, \phi_{\mathcal{P}}) := \mathsf{PoSW}^H(\chi, N)$, where ϕ is a proof and $\phi_{\mathcal{P}}$ is a state $\mathcal{P}$ uses to compute the opening.
- *Opening Challenge:* The opening challenge γ is determined by $\gamma := H(\chi, \phi) \in \{0,1\}^w$.
- *Open:* $\mathcal{P}$ computes $\tau := \mathsf{open}^H(\chi, N, \phi_{\mathcal{P}}, \gamma)$. $\mathcal{P}$ sends $\pi := (\phi, \tau)$ to $\mathcal{V}$.
- *Verify:* $\mathcal{V}$ computes and outputs $\mathsf{verify}(\chi, \mathsf{N}, \phi, \gamma, \tau) \in \{\mathsf{accept}, \mathsf{reject}\}$.

Since our goal is to analyze post-quantum security of Simple PoSW [8], we will not present the formal security properties for PoSW here. Instead, we will prove concrete upper bounds on the probability that a k-parallel q-query quantum oracle algorithm $\mathcal{A}$ with $q < N$ can generate a valid proof.

[20] The original paper [8] considers $\mathcal{X} = \{0,1\}^*$; however, we want $\mathcal{X}$ to be finite so that our results from the previous sections apply. Thus, we simply choose B large enough, so that the scheme is well defined, but also larger than any query that an arbitrary but fixed attacker will make.

C The Extraction Algorithm

Algorithm 1 $\mathsf{Extract}_n^D(\ell_{\mathsf{rt}})$

Input: $\ell_{\mathsf{rt}} \in \{0,1\}^w$
Output: a subtree $T \subseteq V_n$
Initialize:
Set $\ell^{\mathsf{ext}} : V_n \to \{0,1\}^w \cup \{\perp\}$ with $\ell^{\mathsf{ext}}_{\mathsf{rt}} \leftarrow \ell_{\mathsf{rt}}$ and $\ell^{\mathsf{ext}}_v \leftarrow \perp$ for all $v \in V_n \setminus \{\mathsf{rt}\}$;
Set all vertex $v \in V_n$ as unmarked;
Notation: Define the support of a labelling as $\mathsf{Supp}(\ell^{\mathsf{ext}}) := \{v \in V_n : \ell^{\mathsf{ext}}_v \neq \perp\}$
Labelling extraction:
while *there is an unmarked* $v \in \mathsf{Supp}(\ell^{\mathsf{ext}}) \setminus \mathsf{leaves}(V_n)$ **do**
  mark the vertex v;
  if *there exists some* $x, y \in \{0,1\}^w$ *such that* $\ell^{\mathsf{ext}}_v = D(v, x, y)$ **then**
    $\ell^{\mathsf{ext}}_{\mathsf{left}(v)} \leftarrow x$;
    $\ell^{\mathsf{ext}}_{\mathsf{right}(v)} \leftarrow y$;
  end
end
Consistency check:
$T \leftarrow \mathsf{Supp}(\ell^{\mathsf{ext}})$;
for $v \in \mathsf{leaves}(T)$ **do**
  if $\ell^{\mathsf{ext}}_v \neq D(v, \ell^{\mathsf{ext}}_{\mathsf{in}(v)})$ **then**
    $T \leftarrow T \setminus \{v\}$;
  end
end
output T;