Demystifying the Master Thesis and Research in General: The Story of Some Master Theses

Oded Goldreich
Department of Mathematics and Computer Science
Weizmann Institute of Science, Rehovot, Israel.
oded.goldreich@weizmann.ac.il

Revised November 6, 2009

Abstract

I don't think that there is a generically good way of doing a master thesis (nor of doing research in general). Research (like other creative activities) evolves in unpredictable ways, and each research project has its own story. I will tell a few such stories, and naturally I will rely on stories I know from the inside (or close to that). The only lesson that I can offer is maintaining an openness towards ideas that may emerge.

(Notes for a talk to be given on January 20, 2008.)

1 My own thesis (1981)

In one of my first meeting with my predetermined interim supervisor, Shimon Even (who has later become my Master and Doctorate thesis advisor) tossed to my direction a Rubic Cube and asked if I can arrange it. A few days later, when I described to him a highly wasteful algorithm, the question of the minimum move sequence arose naturally. Phrased in general terms, this yields a computational problem regarding permutation groups, to be described next.

A permutation group over a set D is represented by a set of generators; that is, the group generated by a set S of permutation (over D) is defined as

$$\langle S\rangle \stackrel{\rm def}{=} \{g_1 \circ g_2 \circ \cdots \circ g_\ell : g_1, g_2, ..., g_\ell \in S\}$$

where $\circ$ denotes the composition of permutations. For example, in the case of the Rubic Cube, the set of generators corresponds to the 6×2 rotations that can be applied to the cube (where each rotation is determined by a rotating side of the cube and a direction of rotation).

A shortest move sequence between two permutations $\pi_1, \pi_2 \in \langle S\rangle$ is the shortest sequence $(g_1, g_2, ..., g_\ell)$ over S such that $\pi_2 = g_\ell \circ \cdots \circ g_2 \circ g_1 \circ \pi_1$. A natural computational problem is finding, given S and $\pi_1, \pi_2 \in \langle S\rangle$, a shortest move sequence from π_1 to π_2. A computationally equivalent problem refers to finding the shortest sequence of permutations that generates a given permutation. A corresponding decision problem is presented next.

Definition 1 (short generating sequence): *Given a set of generators S, a permutation $\pi \in \langle S\rangle$, and in integer ℓ presented in unary, determine whether or not there exists $\ell' \leq \ell$ and $g_1, g_2, ..., g_{\ell'} \in S$ such that $\pi = g_{\ell'} \circ \cdots \circ g_2 \circ g_1$.*

It is quite easy to show that the foregoing problem is NP-complete, where the unary presentation of ℓ seems essential for the problem being in $\mathcal{NP}$ (since, as shown later, for ℓ presented in binary the problem is actually PSPACE-complete).

My proof of NP-completeness consisted of a simple reduction from 3XC. Recall that an instance of the latter is a sequence of 3-sets over some universe $[3n]$ and the question is whether there exists a subsequence that forms an exact cover of $[3n]$. The reduction maps such an instance to a sequence of generating permutations over $3n$ pairs of elements such that in the $i^{\rm th}$ generator permutation the $j^{\rm th}$ pair is switched if and only if the $i^{\rm th}$ subset contain the element j. The target permutation has all $3n$ pairs switched, and the target length is set to n.

Epilogue: Although the foregoing proof could pass as a Master Thesis in 1981 (but probably not today...), I actually ended-up submitting a different work as my Master Thesis. That work consisted of a taxonomic study of various edge testing problems for networks, where most of these problems were proved to be NP-complete.

2 The thesis of Ronen Vainish (1988)

Ronen was the first master student that I advised. Our joint research was aimed at providing a simplification of the general construction of secure multi-party protocols. The following description assumes some basic familiarity with the subject, as provided in [2, Sec. 7.1].

At the time this research was started, the general construction of secure multi-party protocols proceeded by invoking a general construction of secure two-party protocols multiple times. In retrospect, the most important part of our study is a couple of observations that allow to replace the invocation of the general construction of a secure two-party protocol by a simple protocol.

The first simplifying observation was that the task of constructing arbitrary secure multi-party protocols reduces to providing a secure implementation of the following two-party randomized functionality (for the special case of $n = 2$). For parties holding inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$, respectively, the desired output is a random pair of bits (each obtained at one of the two parties) that sum-up (mod 2) to the inner-product (mod 2) of x and y. In fact, it suffices to consider security in the semi-honest model, where each party follows the prescribed protocol and the question is what can be learned from the full transcript of the party's view of the protocol's execution.

The second simplifying observation was that securely implementing the aforementioned two-party functionality reduces to implementing 1-out-of-2 Oblivious Transfer, OT_1^2, which allows a receiver to obtain one out of two bits held by the sender without letting the sender know which bit was obtained. Following is the implementation suggested for the "inner-product functionality":

Construction 2 *For $i = 1, ..., n$, the first party selects uniformly $c_i \in \{0,1\}$, and invokes OT_1^2 as a sender while providing c_i as its first secret and $c_i + x_i \bmod 2$ as its second secret, and the other party asks for the first secret if and only if $y_i = 1$. Note that, in the $i^{\rm th}$ iteration, the second party obtained the value $c'_i \leftarrow c_i + x_i y_i \bmod 2$. The first party* (locally) *outputs $\sum_{i=1}^n c_i \bmod 2$, whereas the second party* (locally) *outputs $\sum_{i=1}^n c'_i \bmod 2$, and indeed*

$$\sum_{i=1}^n c_i + \sum_{i=1}^n c'_i \;\equiv\; \sum_{i=1}^n (c_i + c'_i) \;\equiv\; \sum_{i=1}^n x_i y_i \pmod 2.$$

3 The thesis of Eyal Kushilevitz (1989)

The thesis of Eyal refers to the notion of perfect zero-knowledge, which seems much more strict than the standard notion of zero-knowledge (see [1, Chap. 4]). The corresponding classes of sets having zero-knowledge and perfect zero-knowledge proofs are denoted $\mathcal{ZK}$ and $\mathcal{PZK}$, respectively.

At the time it was known that the existence of one-way functions implies that $\mathcal{NP} \subseteq \mathcal{ZK}$. In contrast, it was known that $\mathcal{PZK} \subseteq \mathcal{SZK} \subseteq \mathcal{AM} \cap \mathrm{co}\mathcal{AM}$, which implies that it is unlikely that $\mathcal{NP}$ is contained in $\mathcal{PZK}$. Some indications that $\mathcal{PZK}$ may extend beyond $\mathcal{BPP}$ were known, assuming the intractability of either Graph Isomorphism or Quadratic Resideousity (since the corresponding sets were known to be in $\mathcal{PZK}$). But both these assumptions seemed less reliable than the intractability of either factoring or the Discrete Logarithm Problem (DLP). Indeed, the open problem that I offered to Eyal was to provide more reliable evidence to the conjecture $\mathcal{PZK} \neq \mathcal{BPP}$, which he did.

Theorem 3 (Eyal's thesis): *There exists a promise problem in PZK that is computationally equivalent to DLP.*

Thus, assuming that DLP is intractable, PZK must extend beyond (the promise problem version of) BPP.

Interestingly, proving that $\mathcal{PZK}$ extends beyond $\mathcal{BPP}$, based on the conjectured intractability of factoring (or even a more general assumption) is still an open problem.

4 The thesis of Ran Canetti (1992)

So far I told the stories of one thesis emerging from a game, one thesis emerging out of studying a famous result, and one thesis addressing a known open problem. The following story is one of a thesis that emerged from wondering about some material learned in a course.

Taking a course on communication complexity, Ran learned about the complexity gap between deterministic and randomized protocols, and wondered whether there exists a trade-off between the amount of randomness and communication complexity. The answer turned out to be affirmative, and detailing it was the contents of Ran's thesis. Below, I will only outline the gap as taught to Ran.

The setting for communication complexity consists of two parties and a predetermined function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. The first party is given a string $x \in \{0,1\}^n$, the second party is given a string $y \in \{0,1\}^n$, and their goal is to obtain the value $f(x,y)$. We are only interested in the number of bits exchanged between the two parties towards their goal, and totally disregard their local computation time. Clearly, each such function can be computed by exchanging n bits (e.g., the first party sends x to the second party). A complexity gap between deterministic and randomized protocols was known to exist for the equality function (i.e., $\mathsf{eq}(x,y) = 1$ if and only if $x = y$):

- Any deterministic protocol for equality has communication complexity at least n.
- There exists a probabilistic protocol for equality that has error probability $1/3$ and communication complexity $O(\log n)$.

Following are two out of several protocols that may be used to establish the probabilistic communication complexity upper-bound.

Construction 4 (two known probabilistic protocols for the function eq):

1. *Using a good error-correcting code $C : \{0,1\}^n \to \{0,1\}^m$, the first party uniformly selects $i \in [m]$ and sends $(i, C(x)_i)$ to the second party, which outputs 1 if and only if the bit $C(x)_i$ equals the value $C(y)_i$.*

   Note that for $x \neq y$, it holds that $B_{x,y} \stackrel{\text{def}}{=} \{i \in [m] : C(x)_i = C(y)_i\}$ has cardinality at most $m - d$, where d denotes the distance of C.

2. *In this case the inputs x and y are viewed as elements of $\{0, 1..., 2^n - 1\}$. The first party uniformly selects a prime $p \in [n^2, 2n^2]$ and sends $(p, x \bmod p)$ to the second party, which outputs 1 if and only if the value $x \bmod p$ equals the value $y \bmod p$.*

   Using the Chinese Reminder Theorem, for any $x \neq y$, the set of primes $p \in [n^2, 2n^2]$ that satisfy $x \bmod p = y \bmod p$ has cardinality smaller than $n/\log n$.

5 The thesis of Iftach Haitner (2004)

When writing [2], I realized that the standard Oblivious Transfer protocol works under more strict conditioned than commonly assumed. Specifically, I refer to the following protocol (see [2, Sec. 7.3.2] for further details).

Construction 5 (Oblivious Transfer (OT_1^2) protocol for semi-honest model): *The protocol refers to a collection of trapdoor permutation, $\{f_\alpha : D_\alpha \to D_\alpha\}_{\alpha \in I}$, where $D_\alpha \subseteq \{0,1\}^{|\alpha|}$, and to a corresponding hard-core predicate $b : \{0,1\}^* \to \{0,1\}$.*

Inputs: *The sender has input $(\sigma_1, \sigma_2) \in \{0,1\}^2$, the receiver has input $i \in \{1,2\}$.*

Step S1: *The sender uniformly selects an index-trapdoor pair, (α, t), by running the generation algorithm of the said collection, and sends the index α to the receiver.*

Step R1: *The receiver uniformly and independently selects $x_i, y_{3-i} \in D_\alpha$, sets $y_i = f_\alpha(x_i)$, and sends (y_1, y_2) to the sender.*

Step S2: *Upon receiving (y_1, y_2), using the inverting-with-trapdoor algorithm and the trapdoor t, the sender computes $z_j = f_\alpha^{-1}(y_j)$, for both $j \in \{1,2\}$, and sends $(\sigma_1 \oplus b(z_1), \sigma_2 \oplus b(z_2))$ to the receiver.*

Step R2: *Upon receiving (c_1, c_2), the receiver locally outputs $c_i \oplus b(x_i)$.*

The security of the foregoing protocol relies on the assumption that it is possible to uniformly select $y_{3-i} \in D_\alpha$ without knowing $f_\alpha^{-1}(y_{3-i})$ (or making the task of finding this value easy). This assumption clearly holds in case $D_\alpha = \{0,1\}^{|\alpha|}$, and can be proved for some popular candidate collections of trapdoor permutations (see [2, Apdx. C.1] for details). However, I wanted to regain the claim that OT_1^2 can be securely implemented based on any collection of trapdoor permutations, and posed this challenge to Iftach.

Although Iftach did not resolve this challenge, he made significant progress on it. Specifically, he showed that an alternative protocol (indeed a more complicated version of Construction 5) works when using any collection of trapdoor permutations for which D_α has a noticeable density in $\{0,1\}^{|\alpha|}$. It follows that OT_1^2 can be securely implemented based on any such collection (i.e., of trapdoors with "dense" domain). The question of securely implement OT_1^2 based on an arbitrary collection of trapdoor permutations remains open.

6 Brief comments on four recent theses

6.1 Or Sheffet (Dec. 2006)

The thesis (see also [5]) initiates a study of the randomness-complexity of property testing, presenting both general existential bounds and specific efficient algorithms for the case of Bipartiteness. This starting point of the study is the essential role of randomness in property testing, and the focus is on maintaining the low query (and time) complexity of the tester while decreasing its randomness complexity as much as possible.

6.2 Kfir Barhum (Feb. 2007)

The thesis presents fast algorithms for approximating the average distance between pairs of points in a Euclidean space. A follow-up paper [3] confronts the algorithm presented in the thesis with a straightforward algorithm that merely samples pairs of points, and studies the derandomization of the latter algorithm. That is, the question is of constructing a fixed sparse set of pairs that approximates all pairwise distances for any (corresponding) set of points in a Euclidean space (and more generally in any metric space).

6.3 Or Meir (Oct. 2007)

The thesis (see also [6]) is a technical *tour de force* presenting a combinatorial construction of locally testable codes. Loosely speaking, a code is locally testable if it admit a codeword test that probe the string in a constant number of (randomly selected) locations. Or's construction meets the best known parameters, but does in a way that is different and more pleasing than prior constructions. Specifically, it neither rely on sophisticated algebraic constructions nor on a PCP construction.

6.4 Lidor Avigad (Nov. 2009)

This thesis presents a significant extension of the study of the "lowest complexity level" of testing graph properties (in the adjacency representation model). By the "lowest complexity level" I refer to properties that can be tested by a non-adaptive tester of query complexity that is inversely proportional to the proximity parameter. This class was shown in [4, Sec. 6] to contain, for any constant c, the set of graphs that consist of up to c isolated cliques. Looking at the complement graphs, this means that the propert associated with c is being a "blow-up" of the graph consisting of c isolated vertices. Lidor's extension refers to all properties that correspond to being a blow-up of any fixed graph.

References

[1] O. Goldreich. *Foundation of Cryptography – Basic Tools.* Cambridge University Press, 2001.

[2] O. Goldreich. *Foundation of Cryptography: Basic Applications.* Cambridge University Press, 2004.

[3] K. Barhum, O. Goldreich and A. Shraibman. On approximating the average distance between points. In the proceedings of *11th RANDOM*, Springer LNCS, Vol. 4627, pages 509–524, 2007.

[4] O. Goldreich and D. Ron. Algorithmic Aspects of Property Testing in the Dense Graphs Model. *ECCC*, TR08-039, 2008.

[5] O. Goldreich and O. Sheffet. On the randomness complexity of property testing. In the proceedings of *11th RANDOM*, Springer LNCS, Vol. 4627, pages 296–310, 2007.

[6] O. Meir. Combinatorial construction of locally testable codes. *ECCC*, TR07-115, 2007.