Position-based Quantum Cryptography and Catalytic Computation

FLORIAN SPEELMAN

Position-based Quantum Cryptography and Catalytic Computation

Florian Speelman

Position-based Quantum Cryptography and Catalytic Computation

ILLC Dissertation Series DS-2016-08

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: `illc@uva.nl`
homepage: `http://www.illc.uva.nl/`

The investigations were supported by the DIAMANT project, subsidized by the Netherlands Organization for Scientific Research (NWO), the EU projects SIQS and QALGO, and QuSoft.

CWI
Centrum Wiskunde & Informatica

QuSoft
Research Center for Quantum Software

Copyright © 2016 by Florian Speelman

Printed and bound by Ipskamp Drukkers.

ISBN: 978–94–028–0345–7

Position-based Quantum Cryptography and Catalytic Computation

Academisch Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel
op woensdag 16 november 2016, te 12.00 uur

door

Florian Speelman

geboren te Ouder-Amstel.

Promotor:	Prof.dr. H. Buhrman	Universiteit van Amsterdam
Overige leden:	Prof. dr. A. Kent	University of Cambridge
	Dr. C. Schaffner	Universiteit van Amsterdam
	Prof. dr. C.J.M. Schoutens	Universiteit van Amsterdam
	Dr. L. Torenvliet	Universiteit van Amsterdam
	Prof. dr. R. de Wolf	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The results in this thesis are based on the following articles. For all articles, the authors are ordered alphabetically and co-authorship is shared equally.

1. [BFSS13] Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. The garden-hose model. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 145–158, New York, NY, USA, 2013. ACM.

2. [Spe16] Florian Speelman. Instantaneous non-local computation of low T-depth quantum circuits. In *11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, pages 9:1–9:24, 2016.

3. [BCK+14] Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: Catalytic space. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 857–866, New York, NY, USA, 2014. ACM.

4. [BKLS16] Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. Catalytic space: non-determinism and hierarchy. In *33rd Symposium on Theoretical Aspects of Computer Science* (STACS 2016).

The author has additionally (co-)authored the following articles that are not included in this thesis.

5. [BBL+15] Jop Briët, Harry Buhrman, Debbie Leung, Teresa Piovesan, and Florian Speelman. Round elimination in exact communication complexity. In *10th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2015).*

6. [BCG+16] Harry Buhrman, Łukasz Czekaj, Andrzej Grudka, Michał Horodecki, Paweł Horodecki, Marcin Markiewicz, Florian Speelman, and Sergii Strelchuk. Quantum communication complexity advantage implies violation of a Bell inequality. In *PNAS*, 113 (12) 3191–3196, 2016.

7. [BBK+13] [BBK+16] Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolay Vereshchagin.
Towards a Reverse Newman's Theorem in Interactive Information Complexity. In *IEEE Conference on Computational Complexity (CCC 2013)* and *Algorithmica, p. 1–33, 12 January 2016.*

8. [DSS16] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Advances in Cryptology – CRYPTO 2016*, part III p. 3–32, 2016.

Contents

Acknowledgments **xi**

1 Introduction **1**
- 1.1 Position-based quantum cryptography 1
  - 1.1.1 Example: the $\mathrm{QPV}_{\mathrm{BB84}}$ scheme 4
  - 1.1.2 Our contributions 6
- 1.2 Catalytic computation 9

2 Preliminaries **13**
- 2.1 Notation 13
- 2.2 Quantum information 14
  - 2.2.1 Teleportation 18
  - 2.2.2 Mixed states and density matrices 20
  - 2.2.3 The No-Cloning Theorem 21
- 2.3 Communication complexity 21
- 2.4 Complexity theory 23

I Position-based quantum cryptography 25

3 The garden-hose model **27**
- 3.1 Introduction 28
- 3.2 A scheme for position-verification 31
- 3.3 The garden-hose model 33
  - 3.3.1 Definition 33
  - 3.3.2 Upper and lower bounds 35
  - 3.3.3 Equality 37
  - 3.3.4 Inner product 38
  - 3.3.5 Lower bounds 39

3.3.6 Garden-hose complexity and log-space computations . . . 42
3.4 Randomized garden-hose complexity 46
3.5 Quantum garden-hose complexity 48
3.5.1 Deterministic setting 48
3.5.2 Randomized setting 50
3.6 Lower bounds on quantum resources to perfectly attack $\mathsf{PV}_{\text{qubit}}$ . . 51
3.6.1 Localized qubits 52
3.6.2 Squeezing many vectors in a small space 53
3.6.3 The lower bound 54
3.6.4 Functions for which perfect attacks need a large space . . . 56
3.7 Conclusion and open questions 57

4 INQC of low T-depth quantum circuits 59
4.1 Introduction 60
4.2 Preliminaries 63
4.2.1 The Pauli matrices and the Clifford group 63
4.2.2 Key transformations from Clifford circuits 64
4.2.3 Clifford+T quantum circuits, T-count and T-depth 65
4.2.4 The garden-hose model 66
4.3 Definition of INQC 68
4.4 Low T-count quantum circuits 70
4.4.1 The Clifford hierarchy 73
4.5 Conditional application of phase gate using garden-hose protocol . 74
4.6 Low T-depth quantum circuits 78
4.7 Attack on the Interleaved Product protocol 81
4.8 Discussion 86

5 Experimental considerations for single-qubit position verification 89
5.1 Introduction 89
5.1.1 Results 93
5.1.2 Related work 93
5.1.3 Security model for limited communication speed 94
5.1.4 Other protocol modifications 96
5.2 Attack model and proof strategy 96
5.3 Bound by SDP 98
5.3.1 SDP relaxation of monogamy game 98
5.3.2 Deriving the constraints 101
5.3.3 Proof of Lemma 5.3.1 103

II Catalytic computation 105

6 Catalytic computation **107**
6.1 Introduction 108
6.2 Preliminaries 109
6.3 Transparent computation 112
  6.3.1 Previous results on this model 114
  6.3.2 Getting more 115
  6.3.3 Getting TC^1 117
6.4 Catalytic computation 120
  6.4.1 Simulation of transparent computation 121
  6.4.2 Upper bounds 123
  6.4.3 Oracle results for catalytic computation 125

7 Catalytic computation: Non-determinism and hierarchy **129**
7.1 Introduction 129
7.2 Preliminaries 131
  7.2.1 Existence of hash family 133
7.3 Non-deterministic catalytic computation 134
  7.3.1 Simulation by probabilistic computation 136
7.4 An analogue of the Immerman–Szelepcsényi theorem 137
7.5 Hierarchies for Catalytic Computation 145
7.A CNL definition, equivalent to Definition 7.3.2 147

Bibliography **149**

Index **163**

Abstract **167**

Samenvatting **171**

Acknowledgments

I would first and foremost like to thank my advisor, Harry Buhrman. Being his student these past years have been a wonderful experience, and I am very grateful for the opportunity to be a part of his research group at CWI, for his guidance, and for sense of humor. It was always a great pleasure to work together and to learn from his many insights.

For agreeing to be a part of my PhD committee and for their helpful comments on earlier drafts of this thesis, I thank Adrian Kent, Christian Schaffner, Kareljan Schoutens, Leen Torenvliet, and Ronald de Wolf.

Of course this thesis would not be possible without the co-authors of the papers that the chapters are based on, and I am very grateful to Harry Buhrman, Richard Cleve, Serge Fehr, Michal Koucký, Bruno Loff, Christian Schaffner, and Hugo Zbinden. Besides these, I would also like to thank Jop Briët, Joshua Brody, Łukasz Czekaj, Yfke Dulek, Andrzej Grudka, Michał Horodecki, Paweł Horodecki, Debbie Leung, Marcin Markiewicz, Teresa Piovesan, Sergii Strelchuk, and Nikolay Vereshchagin for working together on papers that are outside the scope of this thesis.

QuSoft and the Algorithms & Complexity group were a great environment to be in, and responsible are the colleagues I have had the pleasure to interact with over the past years, who I all want to thank for their good company, ideas, and games of table football – including Joran van Apeldoorn, Srinivasan Arunachalam, Tom Bannink, Ralph Bottesch, Jop Briët, Sabine Burgdorf, André Chailloux, Yfke Dulek, David García Soriano, András Gilyén, Koen Groenland, Peter van der Gulik, Bruno Loff, Fernando de Melo, Teresa Piovesan, Giannicola Scarpa, Christian Schaffner, Penghui Yao, and Jeroen Zuiddam. Of these I'd like to highlight Christian for his invaluable advice and encouragement, Ronald for his very helpful eye for improvements and for kindly sharing his newspaper with me, and the office mates that I have had: Jop, David, Peter, Bruno, and Jeroen. I also thank the members of the Machine Learning group, for their company at many pleasurable lunches.

The quantum information and complexity theory communities have been very welcoming and I have enjoyed my interactions with many researchers. I'd like to thank Richard Cleve, Nicolas Gisin, Adrian Kent, Arie Matsliah, Sergii Strelchuk, and Hugo Zbinden for their hospitality when hosting me, and additionally I'd like to thank Anne Broadbent, Lance Fortnow, Stacey Jeffery, Anthony Leverrier, Periklis Papakonstantinou, and Dominik Scheder for interesting scientific discussions and advice.

Finally, I would like to thank my family, girlfriend, and friends for making these past few years very happy ones.

Amsterdam Florian Speelman
September, 2016.

Chapter 1

Introduction

1.1 Position-based quantum cryptography

The first part of this thesis focuses on position-based quantum cryptography. Most classical cryptography is based on secret keys, but the aim of position-based cryptography is to use position as a credential instead, for example to create messages that are guaranteed to come from a certain location.

The field of quantum information investigates what computational tasks are possible when, instead of ordinary bits, information is stored in quantum-mechanical systems, called qubits. Manipulating qubits makes it possible to use phenomena unique to the laws of quantum mechanics, such as entanglement: the possibility of different particles to be more strongly correlated than possible in a classical theory.

Since its beginnings, the development of quantum computation has been intimately tied to cryptography. The field gained much in prominence when Peter Shor showed in 1994 [Sho94] that factorization of large numbers can be done efficiently by quantum computers, since that implies that the creation of a working quantum computer would break RSA – a widely used public-key cryptosystem. Even though this seems to be bad news for our security, quantum information has also been the source of new cryptography. For example, the BB84 cryptosystem [BB84] generates keys that are provably secure. The BB84 protocol was a major milestone in the field, and besides the theoretical importance of this work, implementations of the scheme are commercially available.

The goal of *position-based cryptography* is to perform cryptographic tasks using location as a credential. Think for example of a scheme that encrypts a message in such a way that this message can only be read at a certain location, like a military base. Position authentication is another example of a position-based cryptographic task; there are many thinkable scenarios in which it would be very useful to be assured that the sender of a message is indeed at the claimed location.

One of the most basic tasks of position-based cryptography is *position verifi-*

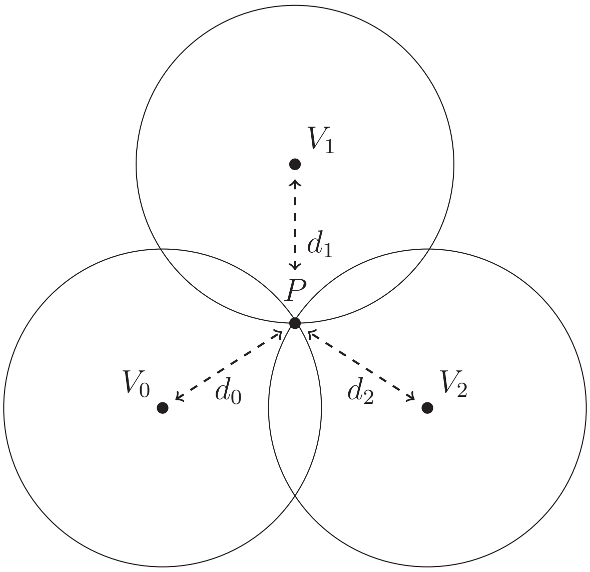

Figure 1.1: Example setup for two-dimensional position verification. The circle centered around the verifiers show the possible locations of any party that can respond to the message in a timely manner. In this picture, P is the only location from where a response can reach all three verifiers in time. A coalition of adversaries will need to use a non-trivial common strategy to break the protocol.

cation. We have a *prover* P trying to convince a set of *verifiers* $V_0, \ldots, V_k$, spread around in space, that P is present at a specific position *pos*. The first idea for such a protocol is a technique called distance bounding [BC94]. Each verifier sends a random string to the prover, using radio or light signals, and measures how long it takes for the prover to respond with this string. Because the signal cannot travel faster than the speed of light, each verifier can upper bound the distance from the prover. For a two-dimensional example, see Figure 1.1.

The current general framework of position-based cryptography was introduced by Chandran, Goyal, Moriarti and Ostrovsky [CGMO09]. Before the recent formulation of a general framework, the problem of secure positioning had been studied in the field of wireless security, and there have been several proposals for this task ([BC94, SSW03, VN04, Bus04, CH05, SP05, ZLFW06, CCS06]).

Although the earlier proposals are provably secure against a single attacker, they can all be broken by *multiple colluding adversaries*. A group of adversaries can send a copy of all information they intercept to their other partners in crime. Each adversary can then emulate the actions of the honest prover and in this way fool the verifier that is closest. It was shown by Chandran et al. [CGMO09] that such an attack is always possible in the classical world, when not making any extra assumptions. Their paper does give a scheme where secure position verification can be achieved, when restricting the adversaries by assuming there is an upper limit to the amount of information they can intercept: the Bounded Retrieval

Model. Assuming bounded retrieval might not be realistic in every setting, so the next question was whether other extensions might be possible to achieve better security.

Attention turned to the idea of using quantum information instead of classical information. Because the classical attacks depend on the ability of the adversaries to simultaneously keep information and send it to all other adversaries, researchers hoped that the impossibility of copying quantum information might make an attack impossible. (See Section 2.2.3 for the quantum no-cloning theorem.)

The first schemes for position-based quantum cryptography were investigated by Kent in 2002 under the name of *quantum tagging*. Together with Munro, Spiller and Beausoleil, a U.S. patent was granted for this protocol in 2006 [KMSB06]. These results have appeared in the scientific literature only in 2010 [KMS11]. This paper considered several different schemes, and also showed attacks on these schemes. Independently in the same year, Malaney proposed schemes that use quantum information for position-verification and location-dependent communication [Mal10a, Mal10b]. Besides these early proposals, multiple other schemes have been put forward, but all eventually turned out to be susceptible to attacks.

Eventually a general impossibility result was given by Buhrman, Chandran, Fehr, Gelles, Goyal, Ostrovsky, and Schaffner [BCF$^+$11], showing that every quantum protocol can be broken. The construction in this general impossibility result uses a doubly exponential amount of entanglement. Beigi and König later gave a new construction, which reduces the needed entanglement to an exponential amount [BK11].

The improved construction by Beigi and König made use of port-based teleportation [IH08, IH09], a novel way of teleporting where the correcting operation of the receiver is very simple (discarding a part of his state), at the cost of using much entanglement. More efficient variants of the protocol have been proposed [SHO13], although these have not yet been applied to position-based quantum cryptography. Port-based teleportation was also used to study the connection between quantum communication complexity and Bell inequalities [BCG$^+$16].

Even though it has been shown that any scheme for position-based quantum cryptography can be broken, these general attacks use an amount of entanglement that is too large for use in practical settings. Even when the honest provers use a small state, the dishonest players need an astronomical amount of EPR pairs to perform the attack described in the impossibility proofs. This brings us to the following question, which is also a central topic of the first part of this thesis:

How much entanglement is needed to break specific schemes for quantum position verification?

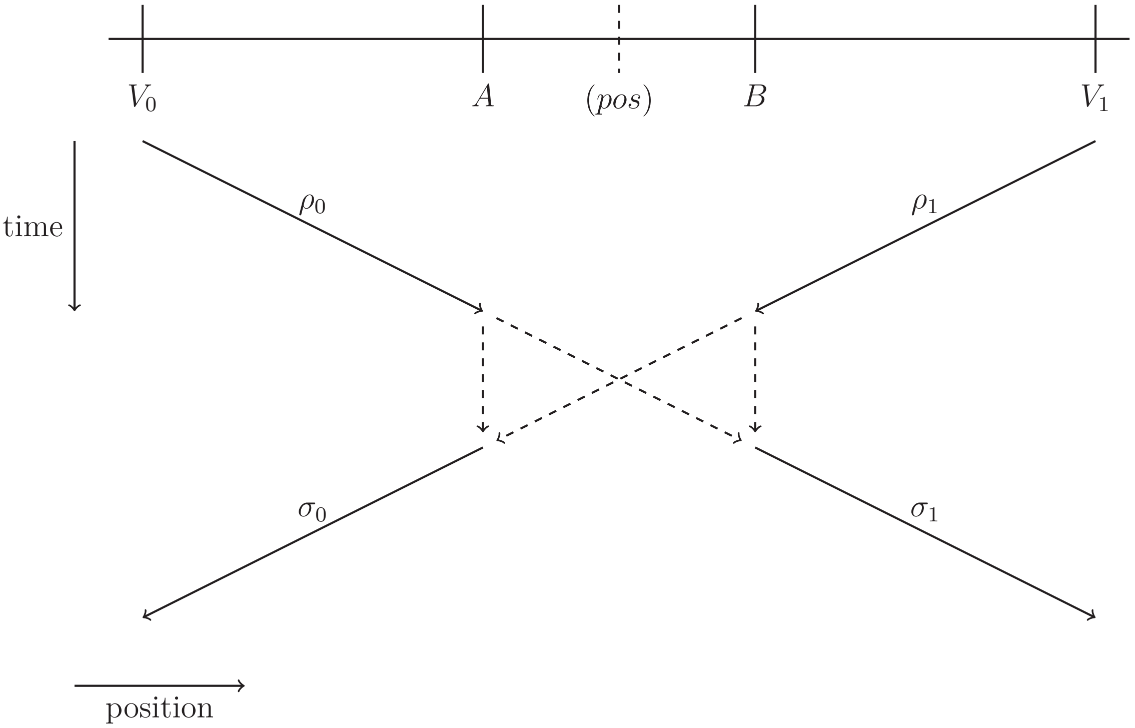

Figure 1.2: Attack on a one-round one-dimensional protocol for position verification, by two attackers Alice and Bob, instead of the (absent) honest prover P at claimed position *pos*. They reply to messages ρ_0 by V_0 and ρ_1 by V_1 with responses σ_0 and σ_1. The attackers have time for one round of simultaneous communication, besides their local quantum memory. Time flows from top to bottom, the horizontal dimension represents position.

1.1.1 Example: the $\mathrm{QPV}_{\mathbf{BB84}}$ scheme

The $\mathrm{QPV}_{\mathrm{BB84}}$ protocol for quantum position verification is the proposal that has currently been studied most. In Figure 1.3 the one-dimensional version has been drawn schematically[1]. The states used are similar to that in the BB84 protocol for quantum key distribution [BB84].

The prover wants to convince the two verifiers, V_0 and V_1, that he is at position *pos* on the line in between them. V_0 sends a qubit $|\phi\rangle$ prepared in one of four states to P: he sends either the states of the computational basis $|0\rangle$ or $|1\rangle$, or the basis states of the Hadamard basis $|+\rangle$ or $|-\rangle$. From the other side V_1 sends the basis θ to P, where we use $+$ to indicate the computational basis and $\times$ to indicate the Hadamard basis. The verifiers V_0 and V_1 time their actions such that the messages arrive at the location of the honest prover at the same time. The prover P has to correctly (and in time) tell V_0 and V_1 which qubit was sent, which

---

[1] For an introduction to the notation used here and to quantum teleportation, see the quantum information preliminaries in Section 2.2.1.

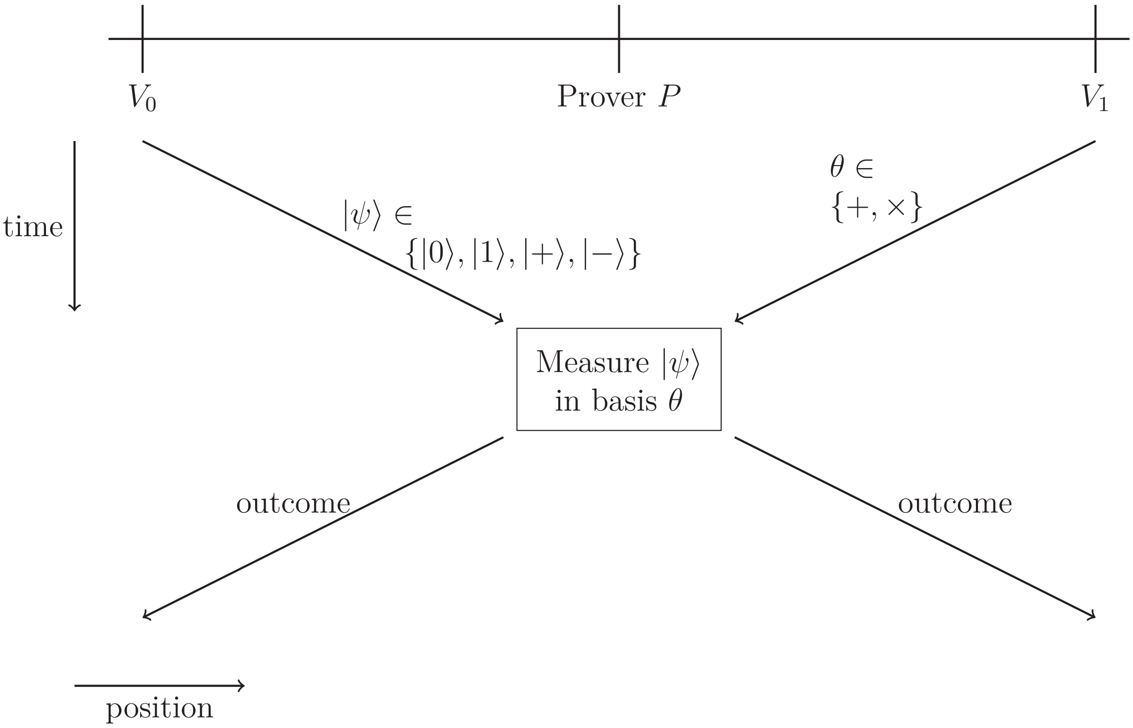
V_0
Prover P
V_1
time
$|\psi\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$
$\theta \in \{+, \times\}$
Measure $|\psi\rangle$ in basis θ
outcome
outcome
position

Figure 1.3: The QPV_{BB84} protocol. The prover receives a quantum state $|\psi\rangle$ from verifier V_0 and a measurement basis θ as classical message from V_1. He has to respond with the measurement outcome to both V_0 and V_1 in time.

he can do by measuring $|\psi\rangle$ in basis θ and immediately broadcasting the outcome.

The work of Buhrman et al. [BCF$^+$11] gave a security proof for this protocol – which holds assuming that attackers, positioned as in Figure 1.1, do not start with an *entangled* quantum state. This result was extended by the work of Tomamichel, Fehr, Kaniewski and Wehner [TFKW13] who show that the entanglement needed grows if the protocol is executed in parallel (the exact bound was later tightened by Ribeiro and Grosshans [RG15])[2].

On the other hand, the QPV_{BB84} protocol can be broken easily by attackers that share entanglement – also see Figure 1.4. The attackers, Alice and Bob[3], only need to share a single EPR pair to perform a successful attack.

The attacker Alice who intercepts the qubit immediately teleports it to Bob, with outcomes the two bits of her teleportation measurement a_1, a_2. The half of

---

[2] The work by Unruh [Unr14] also showed security of a variant of QPV_{BB84}, combined with classical information, but requires existence of a random oracle, a different type of cryptographic assumption than we will consider in this thesis.

[3] Giving attackers the friendly names Alice and Bob is not standard in the literature on quantum cryptography. We choose to use these names, contrary to for example 'multiple eavesdropper Eves' E_0 and E_1, because most of our results are given from the perspective of the attackers, for whom breaking the cryptographic scheme is a cooperative task.

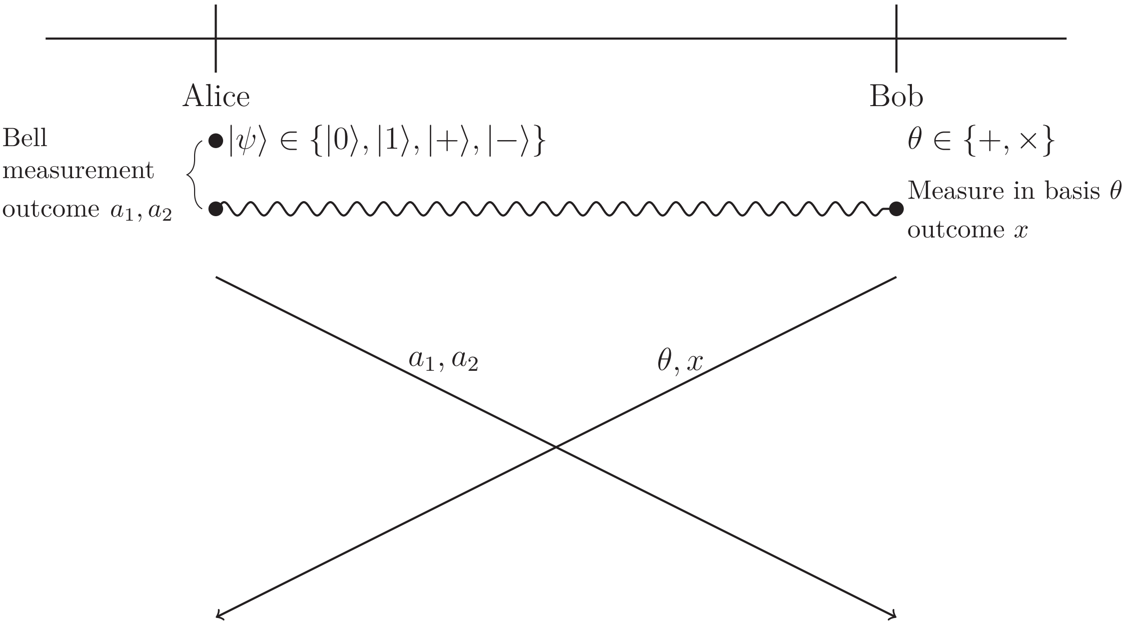
Alice
Bob
Bell measurement outcome a_1, a_2
$|\psi\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$
$\theta \in \{+, \times\}$
Measure in basis θ outcome x
a_1, a_2
θ, x

Figure 1.4: Breaking $\mathrm{QPV}_{\mathrm{BB84}}$, from the perspective of the attackers, Alice and Bob, who share a single EPR pair. Timing constraints force them to use only a single round of simultaneous communication. If $\theta = +$, the players output $x \oplus a_1$. If $\theta = \times$, they output $x \oplus a_2$.

the EPR pair on Bob's side can then be described as $X^{a_1} Z^{a_2} |\psi\rangle$.

Now, if the original qubit $|\psi\rangle$ was in the state $|0\rangle$, the qubit at Bob's side will just be $|0\rangle$ if $a_1 = 0$, and $|1\rangle$ if $a_1 = 1$. These outcomes are precisely opposite if she started with an intercepted $|1\rangle$: when starting with a state in the computational basis, a_1 just determines whether the bit is flipped. Similarly, if $|\psi\rangle$ started as $|+\rangle$ or $|-\rangle$, the state on Bob's side will still be one of $|+\rangle$ or $|-\rangle$, where they are exchanged if $a_2 = 1$.

The other attacker, Bob, has intercepted the basis θ. Simultaneously with Alice's actions, Bob performs the correct measurement given by θ on his half of the EPR pair. After exchanging their measurement results, the attackers now have enough information to produce the correct outcome.

1.1.2 Our contributions

The above scheme uses a quantum state and a single classical bit, but more complicated proposals might be harder to break. In **Chapter 3**, we will investigate several schemes that combine classical and quantum information. Schemes of this form were first considered by Kent et al. [KMS11]. We focus on the one-dimensional set-up, but the schemes easily generalize to three-dimensional space. Besides the assumption that all communication happens at the speed of light, we assume that all parties do not need time to process the verifiers' messages and can perform computations instantaneously. We also assume that the verifiers have clocks that are synchronized and accurate, and that the verifiers have a private channel over which they can coordinate their actions.

The basic scheme we consider can be described as follows. The setup assumes a prover P and two verifiers, V_0 and V_1, with the prover at a position *pos* on the line in between them. V_0 sends a qubit $|\phi\rangle$ prepared in a random basis to P. In addition, V_0 sends a string $x \in \{0,1\}^n$ and V_1 a string $y \in \{0,1\}^n$ to P. The verifiers V_0 and V_1 time their actions such that the messages arrive at the location of the honest prover at the same time. After receiving $|\phi\rangle, x$ and y, P computes a predetermined Boolean function $f(x,y)$. He sends $|\phi\rangle$ to V_0 if $f(x,y) = 0$ and to V_1 otherwise. V_0 and V_1 check that they receive the correct qubit in time corresponding to *pos* and measure the received qubit in the basis corresponding to which it was prepared. In order to cheat the scheme, we imagine two provers P_0 and P_1 on either side of the claimed position *pos*, who try to simulate the correct behavior of an honest P at *pos*.

Looking from the perspective of the adversaries, we can describe their task in the following way. P_0 receives $|\phi\rangle, x$ and P_1 receives y. They are allowed to simultaneously send a single message to each other such that upon receiving that message they both know $f(x,y)$ and if $f(x,y) = 0$ then P_0 still has $|\phi\rangle$, otherwise P_1 has it in his possession. The attack described in [KMS11] accomplishes this task, for any function f, but requires an amount of entanglement that is exponential in n. In this thesis we introduce a complexity measure which relates to the complexity of computing $f(x,y)$, the garden-hose complexity. The garden-hose complexity gives an upper bound on the number of EPR pairs the adversaries need to break the one-qubit scheme that corresponds to the function f.

These protocols are interesting to consider, because the quantum actions of the honest prover are very simple. All the honest prover has to do is route a qubit to the correct location, while the verifiers have to measure in the correct basis, actions which are not much harder than those needed in the BB84 protocol, which is already technologically feasible. If a gap can be shown between the difficulty of the actions of the honest prover and those of the adversaries, this protocol would be a good candidate to investigate further for use in real-life settings. The hope is that for functions $f(x,y)$ that are "complicated enough", the amount of entanglement needed to successfully break the protocol grows at least linearly in the bit length n of the classical strings x, y; we would then require more *classical* computing power of the honest prover, whereas more *quantum* resources are required by the adversary to break the protocol. Since manipulating quantum information is currently orders of magnitude harder than manipulating classical information, such a trade-off is very desirable.

In **Chapter 4** we continue the study of schemes for position-based quantum cryptography and their attacks. For specific proposed protocols efficient attacks are known, for example the work by Lau and Lo [LL11] or the attack on the

$\mathrm{QPV}_{\mathrm{BB84}}$ scheme first mentioned by Kent, Munro and Spiller [KMS11]. Also in that line, the garden-hose model, as described in this thesis, gives efficient attacks for a class of proposed protocols. As described earlier, the attacks on general schemes by Buhrman et al. [BCF$^+$11] and Beigi and König [BK11] use exponentially many resources.

These general attacks use the quantum functionality of the protocol as a black box, executing it without knowledge of its structure. Quantum operations are often described in terms of *quantum circuits*, and the size of quantum circuits corresponds to how difficult a quantum functionality is to implement. Of course, honest parties would like to be able to use a protocol which is not too hard to use in practice.

This gives rise to a natural question: do protocols consisting of simple quantum circuits allow for an efficient attack? We answer this question in the positive, for a specific class of simple circuits.

The circuits we consider consist of gates from the Clifford+T gateset, a common universal set of quantum gates. There are many examples in quantum information where the gates from the Clifford group are easy to implement – see for example work on error-correcting codes [Got98b], classical simulation of quantum circuits [AG04], or cryptographic applications [Chi05, BJ15] – while the non-Clifford gate, in our case the T gate, is hard and needs special care. We show that (single-round) protocols for position-based quantum cryptography based on circuits with a small number of T gates, the *T-count*, or a very small number of layers of T gates, the *T-depth*, can be broken efficiently. As application of our techniques, we also present an attack on a recently-proposed scheme for position verification by Chakraborty and Leverrier [CL15], the Interleaved Product protocol.

These attacks are phrased as protocols for *instantaneous non-local quantum computation*. The attack by Buhrman et al. [BCF$^+$11] was based on a scheme by Vaidman [Vai03] for an *instantaneous non-local measurement*, originally phrased as investigation of the compatibility of special relativity with quantum measurement. That attack and the work by Beigi and König [BK11], extended this task to the more general setting of quantum computation. Besides the application to position-based quantum cryptography, protocols for instantaneous non-local quantum computation can also be applied to other settings, e.g. to reduce communication time in distributed quantum computing.

Our final chapter on position-based quantum cryptography proposes modifications to schemes to make them more suitable for practical implementation. Much of the earlier work tries to show security of schemes for position-based cryptography that require honest parties to have a very low error rate for any round, and assumes that transfer of the quantum state between verifier and prover happens at the speed of light. These requirements are very hard to fulfill with current experimental setups.

For many applications on earth, qubits will have to be implemented as photons in optical fiber cables, and some of these photons will invariably be lost. We therefore need schemes that are secure, even when some rounds have to be aborted because no photon is received by the prover – this significantly helps potential adversaries. A second problem concerns the assumption that all messages travel at the speed of light. Even though classical signals can easily be sent at light speed using radio waves, the speed of light in fiber-optic cable is significantly lower. We address these issues in **Chapter 5** by constructing the new protocol , an extension of earlier protocols which is still secure when taking these experimental constraints into account (against limited adversaries). Our security proof converts attacks on the protocol to strategies for a new variant of a *monogamy of entanglement game*, a non-local game first introduced by Tomamichel, Fehr, Kaniewski and Wehner [TFKW13], originally used to prove security of parallel repetition of the $\mathrm{QPV}_{\mathrm{BB84}}$ scheme among other things. We bound the success probability of this game numerically using *semidefinite programming* (SDP).

1.2 Catalytic computation

The second part of this thesis focuses on (classical) complexity theory—the study of which problems can be solved with a limited amount of technical resources. We will give a short introduction to complexity theory in Section 2.4

The *Turing machine* is one of the most well-studied models of computation. First defined by Alan Turing in the 1936 paper 'On Computable Numbers, with an Application to the Entscheidungsproblem' [Tur36], a Turing machine is a hypothetical device that manipulates symbols on a scratch pad according to a set of rules.

For Turing machines, two resources that are commonly limited are computation time and the space used. Complexity theory tries to order problems in classes based on these resources. For example, P stands for the class of problems that can be solved using by a Turing machine that uses an amount of time which is polynomial in the length of problem. We often say that these problems can be solved *efficiently.* Another well-known class is NP, consisting of the problems for which an answer can be *verified* efficiently, even though finding this answer might be hard to do. For our next topic, we will investigate the setting where the space of the computation is bounded, instead of the time.

Consider the following scenario. Say you want to perform a computation, but only have a small amount of space available. In addition to your small memory, you have an extra bigger hard-drive which is already full, filled with data that you don't need right now, but also do not want to erase. Is it possible to use this extra space, even though the memory is already full?

In **Chapter 6** we introduce the model of *catalytic computation* which formalizes this notion. We show that, surprisingly, it is possible to compute more using this extra full space. The term catalysis comes from chemistry, and describes a reactant which speeds up a chemical reaction but is not consumed – just like the full memory added to our computation.

Problems that can be solved by a computation that only uses an amount of memory logarithmic in the input size are part of the complexity class L, short logarithmic space. For our catalytic computations we focus on the case where we have logarithmic normal working space, augmented with a polynomially-big full tape – a class we call CL for *catalytic log space*. Our main question then becomes: is L the same as CL?

Many programmers will have seen the 'XOR swap': a trick which swaps the contents of two variables, $\mathbf{x}$ and $\mathbf{y}$, with starting contents x and y respectively:

Algorithm 1 XOR swap. Here $\oplus$ stands for the bitwise XOR. The contents of the variable after the statements are shown as comments.

```
$\mathbf{x} \leftarrow \mathbf{x} \oplus \mathbf{y}$        ▷ $\mathbf{x} = x \oplus y$
$\mathbf{y} \leftarrow \mathbf{y} \oplus \mathbf{x}$        ▷ $\mathbf{y} = y \oplus (x \oplus y) = x$
$\mathbf{x} \leftarrow \mathbf{x} \oplus \mathbf{y}$        ▷ $\mathbf{x} = (x \oplus y) \oplus x = y$
```

The technical part of our result builds further on a construction of Ben-Or and Cleve [BC92], who showed surprising power of another very limited model of computation: a machine using only 3 registers, which can be added and multiplied together. Their methods were similar to the XOR swap above, cleverly combining registers and utilizing cancellations to produce the correct answer. By using and extending their construction, we are able to show that the circuit class TC^1 can be computed by the catalytic log-space machines—something which is unlikely to be true for ordinary log-space Turing machines.

In terms of limits to the power of CL, we are able to show that catalytic computations only need polynomial time on average. Therefore, CL has at most the power of the complexity class ZPP, zero-error probabilistic polynomial time.

We continue the study of catalytic computation in **Chapter 7**, by proving equivalent statements of two classical results on space-bounded computation in our new model.

First we extend the model by adding *non-determinism*. For ordinary Turing machines, the complexity class NL, non-deterministic log-space, is the same as its complement coNL, a result known as the Immerman–Szelepcsényi Theorem [Imm88, Sze88]. We show the same for our catalytic versions CNL and coCNL, directly adapting the inductive-counting proof of the Immerman–Szelepcsényi Theorem. There are several obstacles that have to be overcome to make such a proof work. One challenge is that the size of the configuration graph might be exponentially big; we use a *pseudorandom generator* to avoid this problem and search for random seeds where the configuration graph is small. Another issue is that we need to be able to remember and compare different configurations, each taking much space to write down. To solve this problem, we use a hashing algorithm to fingerprint the different configurations.

Finally we present a hierarchy theorem – we show that adding more space enables the catalytic computation to solve strictly more problems. The theorem follows as an adaptation the work of Kinne and van Melkebeek [KvM10], and van Melkebeek and Pervyshev [vMP06], who earlier showed hierarchy theorems for probabilistic and other semantic models of computation.

Chapter 2

Preliminaries

In this chapter we will give some basic definitions and results that will be used in the rest of this thesis. Section 2.1 introduces basic notation and presents a few inequalities that are sometimes needed when discussing the other topics. In Section 2.2 we give some basic results on quantum information that will be used frequently in our results on position-based quantum cryptography. Section 2.3 introduces communication complexity. In Section 2.4 we will give a few basic notions of computational complexity theory.

2.1 Notation

We will use $\mathbb{N}$ for the set of natural numbers. $\mathbb{C}$ denotes the complex numbers. $\mathbb{F}_p$ is the finite field of order p. We often will view bits as elements of $\mathbb{F}_2$, so that addition over the field corresponds to the binary XOR of the bits, and field multiplication corresponds to the binary AND. For any natural number n and finite field $\mathbb{F}$ we use $\mathbb{F}^n$ for the vector space formed by n-tuples of elements of $\mathbb{F}$. For any two elements from these vector spaces x and y, we will always use $x \cdot y$ for their inner product, or dot product, given by $\sum_i x_i y_i$.

Let $k, n \in \mathbb{N}$. We write $[n]$ as a shorthand for the set $\{1, \dots, n\}$. Let $x, y \in \{0,1\}^n$ be bit strings. We write $x \oplus y$ for the bitwise XOR. Sometimes we also use addition $x + y$ for the same operation, when viewing x and y as elements of $\mathbb{F}_2^n$. The *Hamming weight* of a binary string $|x|$ is the number of 1s in the string. The *Hamming distance* between two binary strings x and y, written as $\Delta(x, y)$, is the number of positions where the strings differ; this equals $|x \oplus y|$. For a set S, we use S^* for the set of all arbitrary-length tuples of elements of S. In particular, $\{0,1\}^*$ will be the set of all bit strings.

When written without an explicit base, log always stands for the base-2 logarithm. The natural logarithm is written as ln. We use $\binom{n}{k}$ for the binomial coefficient. When S is a set, $\binom{S}{k}$ is the collection of all k-element subsets of S.

Many statements will hold asymptotically, i.e., when a relevant parameter grows sufficiently large, and it will often be convenient to use big-O notation, which we will briefly review here. Let f, g be two functions from $\mathbb{N}$ to $\mathbb{N}$. The expression $f(n) = O(g(n))$ means that there exists a constant c such that $f(n) \leq c \cdot g(n)$ for every sufficiently large n. We say that $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$, and $f(n) = \Theta(g(n))$ if both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ hold. Finally, the so-called little o is written as $f(n) = o(g(n))$, which is true if for any $\varepsilon > 0$ it holds that $f(n) \leq \varepsilon g(n)$ for every sufficiently large n, and the converse $f(n) = \omega(g(n))$ holds if $g(n) = o(f(n))$.

We will occasionally need to bound sums of random variables, for which the *Chernoff bound* can be a very useful tool. Let $X_1, X_2, \ldots, X_n$ be independent random variables, that each are 1 with probability $1/2 + \epsilon$ and 0 with probability $1/2 - \epsilon$ for a positive real number ϵ. Then the Chernoff bound states that

$$\Pr[\sum_{i=1}^{n} X_i \leq n/2] \leq e^{-2\epsilon^2 n} .$$

Another basic inequality we need is the *Cauchy-Schwarz inequality*. In terms of vectors $|u\rangle, |v\rangle$, it states that $|\langle u|v\rangle|^2 \leq \langle u|u\rangle\langle v|v\rangle$. For the special case concerning vectors x, y over $\mathbb{R}^n$, i.e. two lists of real numbers $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$, we obtain the following useful bound:

$$\left(\sum_{i=1}^{n} x_i y_i\right)^2 \leq \left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i^2\right) .$$

2.2 Quantum information

Here we will give a very short introduction to the parts of quantum information that will be most relevant for this thesis. For an excellent textbook on the topic, see [NC00]. Some background knowledge in linear algebra will be assumed for the next section.

The main object of study in quantum information is the *quantum state*, a mathematical description of a system following the laws of quantum mechanics.

Dirac notation As is common in the study of quantum information, we will typically write vectors in bra-ket notation, also known as Dirac notation. To illustrate, consider the finite-dimensional complex Hilbert space $\mathbb{C}^d$ for some number $d \in \mathbb{N}$. In bra-ket notation we write a column vector named ψ as

$$|\psi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_d \end{pmatrix}$$

which is called a *ket*. The *bra* is then the corresponding row vector,

$$\langle\psi| = \begin{pmatrix} \psi_1^* & \psi_2^* & \dots & \psi_d^* \end{pmatrix},$$

where the * denotes complex conjugation. The inner product of two vectors $|\psi\rangle$ and $|\phi\rangle$ can be written as $\langle\psi|\phi\rangle$, a shorthand for $\langle\psi||\phi\rangle$.

The tensor product (which we will introduce below) between two vectors $|\psi\rangle$ and $|\phi\rangle$ is written as $|\psi\rangle \otimes |\phi\rangle$, often abbreviated as $|\psi\rangle|\phi\rangle$.

State vector Quantum states can be described as unit vectors in a complex *Hilbert space*; a complex vector space with an inner product.

The quantum states we will deal with, will mostly be finite-dimensional systems, and therefore our space will often be $\mathbb{C}^d$ for some d. A quantum mechanical state in some Hilbert space $\mathcal{H}$ can be described by a vector $\psi \in \mathcal{H}$, or $|\psi\rangle$ in braket notation. The *vector norm* of a vector ψ, sometimes called the *length* of the vector, is written as $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$. For an operator A we will also sometimes need the *operator norm*, defined in the following way, using the vector norm:

$$\|A\| = \sup_{\psi:\|\psi\|=1} \|A|\psi\rangle\|$$

An important example of a system is a quantum bit or *qubit*. The state of a single qubit is described by a vector in a two-dimensional state space, $\mathcal{C}^2$. The most common orthonormal basis we use for qubits is called the *computational basis* and is defined as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \; |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

We can describe any one-qubit state by a superposition of these basis vectors, enabling us to write

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \text{with} \quad \alpha, \beta \in \mathbb{C}$$

for any one-qubit state $|\psi\rangle$. Here normalization requires that $|\alpha|^2 + |\beta|^2 = 1$. In vector notation we would write this state $|\psi\rangle$, and its dual $\langle\psi|$, as

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \; \langle\psi| = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix}$$

The joint state of multiple quantum systems is a vector in a space that is a *tensor product*[1] of the original spaces. Take as example a Hilbert space V of dimension m and a Hilbert space W of dimension n. The tensor space $V \otimes W$ is a single space of dimension mn, with elements linear combinations of tensor products of the elements of the spaces. For instance, if $|v_1\rangle$ and $|v_2\rangle$ are elements of V, and $|w_1\rangle$ and $|w_2\rangle$ are elements of W, then $|v_1\rangle \otimes |w_1\rangle$ and $|v_2\rangle \otimes |w_2\rangle$ are elements of the tensor space $V \otimes W$, and so is their sum $|v_1\rangle \otimes |w_1\rangle + |v_2\rangle \otimes |w_2\rangle$.

---

[1] See also any textbook on quantum computation, such as [NC00, Section 2.17], for a comprehensive explanation.

Three important properties of the tensor product are the following. For any scalar c, and arbitrary $|v\rangle \in V$ and $|w\rangle \in W$, it holds that

$$c(|v\rangle \otimes |w\rangle) = (c|v\rangle) \otimes |w\rangle = |v\rangle \otimes (c|w\rangle)\,.$$

For any $|v_1\rangle, |v_2\rangle \in V$ and $|w\rangle \in W$,

$$|v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle = (|v_1\rangle + |v_2\rangle) \otimes |w\rangle\,,$$

and similarly for any $|v\rangle \in V$ and $|w_1\rangle, |w_2\rangle \in W$,

$$|v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle = |v\rangle \otimes (|w_1\rangle + |w_2\rangle)\,.$$

For example, a two-qubit system can be described by a unit vector in $\mathbb{C}^4$, with computational basis states

$$|0\rangle|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |0\rangle|1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|1\rangle|0\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad |1\rangle|1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

The state space of n qubits is $\mathbb{C}^{2^n}$, a complex vector space with dimension 2^n. Not all two-qubit states can be written as the tensor product of two one-qubit states. A quantum state that cannot be written as a product of individual qubit states is said to be *entangled*.

A very important two-qubit entangled state is the *EPR pair*. We can write this state in terms of the computational basis states as

$$\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)\,.$$

The evolution of a closed quantum system is described by a *unitary transformation*. This means that we can describe manipulation of the quantum states as a unitary matrix; a matrix for which it holds $U^\dagger U = \mathbb{I}$. Here $A^\dagger$, pronounced as ‘dagger’, is the conjugate transpose of some matrix A, also called Hermitian transpose, obtained by taking the transpose of the matrix and then taking the complex conjugate of each entry.

The *Pauli matrices* are four unitary matrices that are very common in quantum computing. Here we define them as

$$\sigma_0 := \mathbb{I} := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \sigma_1 := \mathrm{X} := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$\sigma_2 := \mathrm{Y} := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_3 := \mathrm{Z} := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} .$$

The following commutation relations between the four Pauli matrices are often helpful to keep in mind:

$$\mathrm{XY} = i\mathrm{Z}$$
$$\mathrm{YZ} = i\mathrm{X}$$
$$\mathrm{ZX} = i\mathrm{Y}$$

When using the Pauli matrices as operations on quantum states the global phase factor i is not important, and will often be omitted.

The *Hadamard matrix* is a unitary transformation which is defined as

$$\mathrm{H} := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

We write

$$|+\rangle := \mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{and}$$
$$|-\rangle := \mathrm{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

for the basis vectors of the *Hadamard basis*.

Measurement Even though our description of a system of n qubits is very big, taking 2^n complex numbers, we can not access these numbers directly. Instead, they give rise to probabilities we will see a certain outcome when we measure this system. A *quantum measurement* is described by a collection $\{M_m\}$ of measurement operators, where m refers to the measurement outcome. If the state before measurement is $|\psi\rangle$, the probability that result m occurs is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle ,$$

and the state after getting measurement outcome m is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}} .$$

Reflecting the fact that probabilities sum to one, we have the *completeness relation*

$$\sum_m M_m^\dagger M_m = \mathbb{I} \, .$$

Measurements in the computational basis can be described by measurement operators

$$M_0 = |0\rangle\langle 0| \qquad\qquad M_1 = |1\rangle\langle 1| \, ,$$

while measurements in the Hadamard basis have measurement operators

$$M_0 = |+\rangle\langle +| \qquad\qquad M_1 = |-\rangle\langle -| \, .$$

Many of the measurement operators we will explicitly use are projective measurements. We call a measurement a *projective measurement* if, beside satisfying the completeness relation, the M_m are orthogonal projectors. The last requirement means that the operators have to be Hermitian, that is $M_m^\dagger = M_m$, and that $M_m M'_m = \delta_{m,m'} M_m$. Here $\delta_{m,m'}$ is the Kronecker delta. As a consequence of using projective measurements, measuring the same qubit twice, consecutively, will give the same outcome both times.

In many cases, when projective measurements are not general enough but when we do not need the measurement operators $\{M_m\}$ to explicitly describe the post-measurement state, it will be convenient to use the *POVM formalism*. Consider a measurement $\{M_m\}$, and suppose we define

$$E_m \equiv M_m^\dagger M_m \, ,$$

then each E_m is a positive semidefinite operator, called a *POVM element*, and we have the relation $\sum_m E_m = \mathbb{I}$. The probability of any particular measurement outcome m is given by $p(m) = \langle\psi|E_m|\psi\rangle$. The set of these operators $\{E_m\}$ is known as a *POVM*. In many circumstances, for example when converting a question in quantum information to an optimization problem, it can be very helpful to argue about a corresponding POVM instead of directly considering the measurement operators.

2.2.1 Teleportation

One of the most important protocols in the study of quantum communication is *quantum teleportation*, first described by Bennett et al. [BBC$^+$93]. The goal of quantum teleportation is to transfer a quantum state from one location to another by only communicating classical information, together with a pre-shared entangled state.

To illustrate, let's say Alice wants to teleport a qubit Q to Bob, in an arbitrary unknown state

$$|\psi\rangle_Q = \alpha|0\rangle_Q + \beta|1\rangle_Q \, .$$

Alice and Bob already share an EPR pair $\frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B$, where Alice has qubit A and Bob has B.

Now consider the following states, the four *Bell states*

$$\begin{array}{ll} |\beta_{00}\rangle := \frac{1}{\sqrt{2}}\Big(|00\rangle + |11\rangle\Big) & |\beta_{10}\rangle := \frac{1}{\sqrt{2}}\Big(|00\rangle - |11\rangle\Big) \\ |\beta_{01}\rangle := \frac{1}{\sqrt{2}}\Big(|01\rangle + |10\rangle\Big) & |\beta_{11}\rangle := \frac{1}{\sqrt{2}}\Big(|01\rangle - |10\rangle\Big) \end{array} . \tag{2.1}$$

Together the four Bell states form an orthonormal basis for the state of two qubits. A *Bell measurement* is a projective measurement in this basis, with the an index of one of the four Bell states as outcome.

For instance, it's easy to check that we can write our original basis vectors in terms of the Bell states in the following way:

$$\begin{array}{ll} |00\rangle = \frac{1}{\sqrt{2}}\Big(|\beta_{00}\rangle + |\beta_{10}\rangle\Big) & |10\rangle = \frac{1}{\sqrt{2}}\Big(|\beta_{01}\rangle - |\beta_{11}\rangle\Big) \\ |01\rangle = \frac{1}{\sqrt{2}}\Big(|\beta_{01}\rangle + |\beta_{11}\rangle\Big) & |11\rangle = \frac{1}{\sqrt{2}}\Big(|\beta_{00}\rangle - |\beta_{10}\rangle\Big) \end{array} . \tag{2.2}$$

The goal of quantum teleportation is to transfer a quantum state from one location to another by only communicating classical information. Teleportation requires pre-shared entanglement among the two locations.

Say Alice wants to teleport a qubit Q to Bob, in an arbitrary unknown state

$$|\psi\rangle_Q = \alpha|0\rangle_Q + \beta|1\rangle_Q \ .$$

The state that Alice and Bob share can also be written $|\beta_{00}\rangle_{AB}$, where Alice has qubit A and Bob has B. The total state of their system then is $|\Psi\rangle = |\beta_{00}\rangle_{AB}|\psi\rangle_Q$.

We can rewrite the state of their quantum system as

$$\begin{aligned} |\Psi\rangle &= \frac{1}{\sqrt{2}}\Big(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B\Big)\Big(\alpha|0\rangle_Q + \beta|1\rangle_Q\Big) \\ &= \frac{1}{\sqrt{2}}\Big(\alpha|00\rangle_{AQ}|0\rangle_B + \beta|01\rangle_{AQ}|0\rangle_B + \alpha|10\rangle_{AQ}|1\rangle_B + \beta|11\rangle_{AQ}|1\rangle_B\Big) \\ &= \frac{1}{2}\Big[\alpha\Big(|\beta_{00}\rangle_{AQ} + |\beta_{10}\rangle_{AQ}\Big)|0\rangle_B + \beta\Big(|\beta_{01}\rangle_{AQ} + |\beta_{11}\rangle_{AQ}\Big)|0\rangle_B + \\ &\qquad \alpha\Big(|\beta_{00}\rangle_{AQ} - |\beta_{10}\rangle_{AQ}\Big)|1\rangle_B + \beta\Big(|\beta_{01}\rangle_{AQ} - |\beta_{11}\rangle_{AQ}\Big)|1\rangle_B\Big] \\ &= \frac{1}{2}\Big[|\beta_{00}\rangle_{AQ}\Big(\alpha|0\rangle_B + \beta|1\rangle_B\Big) + |\beta_{01}\rangle_{AQ}\Big(\alpha|1\rangle_B + \beta|0\rangle_B\Big) \\ &\qquad + |\beta_{10}\rangle_{AQ}\Big(\alpha|0\rangle_B - \beta|1\rangle_B\Big) + |\beta_{11}\rangle_{AQ}\Big(\alpha|1\rangle_B - \beta|0\rangle_B\Big)\Big] \\ &= \frac{1}{2}\Big[|\beta_{00}\rangle_{AQ}\,|\psi\rangle + |\beta_{01}\rangle_{AQ}\,\mathrm{X}|\psi\rangle + |\beta_{10}\rangle_{AQ}\,\mathrm{Z}|\psi\rangle + |\beta_{11}\rangle_{AQ}\,\mathrm{XZ}|\psi\rangle\Big] \ . \end{aligned}$$

Here we reordered terms, and then used Equation 2.2 to write these terms in a different basis – the state that we started with is still unchanged so far. Next Alice performs a Bell measurement on qubits A and Q, getting an outcome $z \in \{00, 01, 10, 11\}$. After this measurement, the state Bob holds will be equal to $\sigma_z|\psi\rangle$, where σ_z is one of the four Pauli matrices that we defined earlier, depending on the outcome z. Now Alice sends the two bits z to Bob.

We can quickly check that when $z = 00$, Bob does not have to apply a correction. On $z = 01$, Bob can recover $|\psi\rangle$ by applying $\sigma_1 = X$. When $z = 10$, Bob has to apply $\sigma_3 = Z$. And when $z = 11$, Bob can recover the original state $|\psi\rangle$ by applying $\sigma_2 = Y$ to his qubit. The Y operation does contain an extra factor i in its usual definition, but this only adds a global phase to the quantum state, which we can always ignore. With this protocol, Alice can effectively transfer a quantum state to Bob, using a pre-shared entangled state and two bits of classical information per qubit.

2.2.2 Mixed states and density matrices

In addition to the description of quantum states with state vectors as above, it is also possible to describe quantum-mechanical systems using *density matrices*. We say that a state that can be represented by a single vector is a *pure state*. The language of density matrices can be especially convenient when describing a system in a *mixed state*.

A mixed state can be defined as a probability distribution over pure states. The resulting mixed state does not always have a unique decomposition as a probability distribution over pure states. For example, consider an experimenter who prepares either $|0\rangle$ with probability 1/2 or $|1\rangle$ with probability 1/2. No measurement can ever distinguish the resulting mixed state from the case where either $(|0\rangle + |1\rangle)/\sqrt{2}$ or $(|0\rangle - |1\rangle)/\sqrt{2}$ is prepared with equal probability.

A density matrix is a positive semidefinite matrix with trace 1 which represents such a mixed state. For a Hilbert space $\mathcal{H}$, we will use $\mathcal{S}(\mathcal{H})$ to denote the set of density matrices that use the corresponding Hilbert space.

For any density matrix there exists a decomposition of the following form, where p_i are non-negative real numbers such that $\sum p_i = 1$ and $|\psi_i\rangle$ are a set of orthonormal vectors.

$$\rho = \sum_i |\psi_i\rangle\langle\psi_i|$$

The collection $\{(p_i, |\psi\rangle)\}$ is called an *ensemble of pure states*. Pure states correspond to rank 1 density matrices; the state ρ can then be written as $|\phi\rangle\langle\phi|$ for some pure state $|\phi\rangle$.

We can write the earlier operations on pure quantum states in terms of density matrices. A unitary transformation U transforms the state ρ to $U\rho U^\dagger$. The probability of getting a certain outcome of a POVM $\{E_m\}$ is given by $\mathrm{tr}(E_m\rho)$.

2.2.3 The No-Cloning Theorem

The *no-cloning theorem* is a classic result of quantum information which states that it is impossible to copy an arbitrary quantum state. This theorem has very important consequences for quantum cryptography. For example, without the impossibility of cloning the BB84 scheme would be insecure. The no-cloning theorem is also the reason why the classical attack on schemes for position-based cryptography does not generalize to the quantum case. Here we state the theorem which shows that perfect cloning is impossible as illustration of the concept: for cryptographic purposes stronger versions which also concern approximate copying are necessary.

2.2.1. THEOREM. *There exists no unitary operation U that perfectly copies the state of an arbitrary qubit.*

Proof. By contradiction, suppose we have a unitary operation U that performs a copy, so that $U|\psi\rangle|s\rangle = U|\psi\rangle|\psi\rangle$ for any possible $|\psi\rangle$, where $|s\rangle$ is some starting state that is independent of $|\psi\rangle$. More specifically, this would imply

$$U(|0\rangle|s\rangle) = |0\rangle|0\rangle$$

and also

$$U(|1\rangle|s\rangle) = |1\rangle|1\rangle.$$

But now let us try to copy $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Since U is linear, we can use the previous equations to get

$$U|+\rangle|s\rangle = \frac{1}{\sqrt{2}}(U|0\rangle|s\rangle + U|1\rangle|s\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$$

and this is not equal to $|+\rangle|+\rangle$, giving a contradiction. □

2.3 Communication complexity

Communication complexity was introduced by Yao [Yao79]. For an excellent introduction to the topic, see the book of Kushilevitz and Nisan [KN97] or one of the more-recent surveys on a specific aspect of communication complexity [LS07, CP10, Raz11].

Alice and Bob want to work together to compute a function $f(x, y)$, where Alice receives the input $x \in \{0, 1\}^n$ and Bob receives $y \in \{0, 1\}^n$. To do this, they are allowed to send messages back-and-forth.

A communication *protocol* defines the messages Alice and Bob send each other. The protocol describes, for possible inputs and previous messages that are sent, the actions for Alice and Bob respectively.

A *transcript* of a protocol is a binary string describing the messages sent by Alice and Bob on a specific run of the protocol.

The *cost* of a protocol is the length of the longest transcript of a protocol, when considered over all input pairs x, y.

The deterministic communication complexity $D(f)$ of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is the lowest cost of any protocol for which Alice and Bob compute $f(x,y)$ correctly on all input pairs.

We can also allow the players to each flip coins, a type of randomness called *private randomness*, and allow the players to be wrong with some small probability. The randomized communication complexity $R_\epsilon(f)$ of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is the lowest cost of any protocol which computes $f(x,y)$ with error at most ϵ for any input pair. We define $R(f)$ as $R_{1/3}$, the cost of the best randomized protocol where the players are allowed to be wrong with probability $1/3$. The specific constant is not important here, since we can reduce the error probability to any desired error ϵ with only a small multiplicative overhead. To do this, we can repeat the protocol $O(\log 1/\epsilon)$ times and then using the outcome that is seen most frequently. The proof then follows directly by applying the Chernoff bound.

Instead of locally flipping coins, we can also imagine the case where the players share *public randomness*, for example by distributing their random choices before they receive their inputs, or because they share a source of external randomness (perhaps both parties have their telescopes pointed towards the same quasar). It turns out that private randomness is capable of simulating public randomness at the cost of a small additive amount of extra error and communication, a result known as *Newman's Theorem* [New91].

Quantum communication complexity After the development of quantum information theory, the model of communication complexity was extended to incorporate quantum information by Yao in 1993 [Yao93]. In the quantum case, Alice and Bob are allowed to communicate qubits, instead of classical bits. For some problems, communicating qubits instead of classical bits can make a very big difference.

The *exact quantum communication complexity* of a function f, denoted by $Q_E(f)$, is the least number of qubits that two players have to communicate to compute the function without any error.

In *bounded-error quantum communication*, we require that the players answer correctly with probability at least $1 - \epsilon$. The bounded-error quantum communication complexity of a function f is called $Q_\epsilon(f)$.

Cleve and Buhrman defined a different variant of quantum communication, one where the parties are only allowed to communicate classical bits, but start the protocol with an (unbounded) entangled quantum state [CB97]. Define the least amount of classical bits that these quantum players have to send to correctly compute a function f as $Q_{\mathrm{CB},\epsilon}(f)$. Using quantum teleportation it is easy to see that for any function, we have $Q_{\mathrm{CB},\epsilon}(f) \leq Q_\epsilon(f)$. It remains an open question to characterize exactly how much more power this entanglement provides, over the model where quantum communication is allowed.

2.4 Complexity theory

Computational complexity theory studies the amount of resources (for example time or space) needed to solve computational problems by a computational model. For a modern textbook on complexity theory, see the recent book by Arora and Barak [AB09]. Here we do introduce a few of the concepts that are important for the final chapters of this thesis.

The *Turing machine* is the first model we will consider, an abstraction of an algorithm which operates on a *memory* according to a set of rules.

The memory of a Turing machine consists of an *input tape*, one or more *work tapes*, and an *output tape*. These tapes consist of a line of cells, each of which contains a symbol from a finite *alphabet*. We will often use an alphabet consisting of just the binary digits 0/1 and 'blank' symbol $\square$, but the exact alphabet size is often not important: Turing machines with smaller alphabets can simulate those with larger ones without a big overhead. Every tape has a *tape head* that can read and write symbols, one cell at a time. At every time step, the tape head can move one cell to the left or to the right.

A Turing machine has a finite set of states, and the *state register* of the machine always contains one of these states. To determine the actions of the Turing machine, the machine has a *transition function* that, given the current state and the symbols under the tape heads, determines the next state in the state register, what to write under the heads, and the movement of the heads.

Computational problems are often given as *decision problems*, i.e., yes or no questions. The most common way to encode such a problem is as a set of arbitrarily long binary strings $A \subseteq \{0,1\}^*$. The computation receives as input some binary string $x \in \{0,1\}^*$, and has to decide whether this string is part of our set or not. For example, if the set would be the set of all prime numbers, the computation has to *accept* if the input x is a prime number and *reject* if the number is composite.

The *time complexity* of a problem is the runtime of the best algorithm that solves it, in terms of the length of the input $|x|$. We say that a problem is *polynomial-time computable* if there exists an algorithm for the problem and a polynomial p such that the algorithm solves the problem in time $p(|x|)$. The class of all polynomial-time computable problems is called P.

Another natural way of classifying how hard a problem is to compute, is in terms of its *space complexity*: the amount of work-tape cells used by a Turing machine solving the problem. Space-bounded computation will feature prominently in the results on the garden-hose model in Chapter 3, and is the central object of study in our work on catalytic computation of Chapter 6 and 7.

A problem A is *log-space computable* if there exists an algorithm that for any input $x \in \{0,1\}^*$ correctly decides whether $x \in A$ or not, and takes space $O(\log |x|)$. The class of all these problems is called L, or sometimes log space.

The Turing machine model can be made more powerful by adding randomness. The class ZPP, short for 'zero-error probabilistic polynomial time', is the set of problems that can be solved without error, in polynomial time on expectation.

ZPP consists of those problems L for which there exists a Turing machine M such that the following holds. Let p be some polynomial. The machine receives, besides its input x, also a polynomially long random string r. The computation has to be correct for any random string r:

$$\forall n, x \in \{0,1\}^n, r \in \{0,1\}^* : M(x,r) = 1 \text{ if and only if } x \in L$$

The runtime does not necessarily have to be polynomial for any random run, we only require that on average the machine halts in polynomial time:

$$\forall x \in \{0,1\}^n \underset{r}{\mathbb{E}}[\text{runtime of } M(x,r)] = p(n)$$

The class NP consists of the problems such that, for any instance that is true, there exists a proof of that fact which can be verified in polynomial time. More precisely, we say that a problem $L \subseteq \{0,1\}^*$ is in NP if there exists a Turing machine M, which runs in time polynomial in its input length, and a polynomial p, such that for any string x:

$$x \in L \iff \exists u \in \{0,1\}^{p(|x|)} M(x,u) \text{ accepts}$$

Some computational problems are at least as difficult as the entire class NP, we call such problems *NP hard*. We say that a problem $S \subseteq \{0,1\}^*$ is NP hard if for any language $L \in \mathsf{NP}$ there exists some function f which is computable in polynomial time, such that for any $x \in \{0,1\}^*$, it holds that

$$x \in L \iff f(x) \in S\,.$$

This means that S is at least as hard as L; any polynomial-time algorithm for S can be turned into an algorithm for L by first using f and then running the algorithm which decides L.

A problem that is both in NP and NP hard is called *NP complete*. The NP-complete problems are the hardest problems in the class NP and an efficient algorithm for any one of them would imply efficient algorithms for all problems in NP. The canonical NP-complete problem is boolean formula satisfiability, SAT, independently proven by Cook [Coo71] and Levin [Lev73]. Many other natural problems are known to be NP complete, such as the traveling salesman problem, the question of whether a graph is colorable with three colors, and more [Kar72, GJ79].

Part I

Position-based quantum cryptography

Chapter 3

The garden-hose model

We define a new model of communication complexity, called the *garden-hose model*. Informally, the garden-hose complexity of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is given by the minimal number of water pipes that need to be shared between two parties, Alice and Bob, in order for them to compute the function f as follows: Alice connects her ends of the pipes in a way that is determined solely by her input $x \in \{0,1\}^n$ and, similarly, Bob connects his ends of the pipes in a way that is determined solely by his input $y \in \{0,1\}^n$. Alice turns on the water tap that she also connected to one of the pipes. Then, the water comes out on Alice's or Bob's side depending on the function value $f(x,y)$.

The garden-hose model was inspired by attacks on a certain class of *quantum position-verification* schemes, where qubits are teleported back-and-forth between cooperating parties. We show an interesting connection between the garden-hose model and the (in)security of these schemes.

We prove almost-linear lower bounds on the garden-hose complexity for concrete functions like inner product, majority, and equality, and we show the existence of functions with exponential garden-hose complexity. Furthermore, we show a connection to classical complexity theory by proving that all functions computable in log-space have polynomial garden-hose complexity.

Finally, we also consider a *randomized* variant of the garden-hose complexity, where Alice and Bob hold pre-shared randomness, and a *quantum* variant, where Alice and Bob hold pre-shared quantum entanglement, and we show that the randomized garden-hose complexity is within a polynomial factor of the deterministic garden-hose complexity. Examples of (partial) functions are given where the quantum garden-hose complexity is logarithmic in n while the classical garden-hose complexity can be lower bounded by n^c for constant $c > 0$.

The results in this chapter are based on the following publication:

- [BFSS13] Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. The garden-hose model. In *Proceedings of the 4th Conference* *on Innovations in Theoretical Computer Science*, ITCS '13, pages 145–158, New York, NY, USA, 2013. ACM.

The conference publication is an extension of the author's earlier Master's thesis:

- [Spe11] Florian Speelman. Position-based quantum cryptography and the garden-hose game. Master's thesis, University of Amsterdam, 2011.

The material on the quantum garden-hose model, mostly concentrated in Section 3.5, was not present in the Master's thesis [Spe11] and was added at the time of conference publication [BFSS13]. Additionally, some of the smaller observations, like Propositions 3.3.10 and 3.3.11, were unpublished before this thesis, the proof of Theorem 3.3.15 has been improved from earlier versions, and references to several later results by other authors have been included.

3.1 Introduction

The garden-hose model On a beautiful sunny day, Alice and Bob relax in their neighboring gardens. It happens that their two gardens share s water pipes, labeled by the numbers $1, 2, \ldots, s$. Each of these water pipes has one loose end in Alice's and the other loose end in Bob's garden. For the fun of it, Alice and Bob play the following game. Alice uses pieces of hose to locally connect some of the pipe ends that are in her garden with each other. For example, she might connect pipe 2 with pipe 5, pipe 4 with pipe 9, *etc.* Similarly, Bob locally connects some of the pipe ends that are in his garden; for instance pipe 1 with pipe 4, *etc.* We note that no T-pieces (nor more complicated constructions), which connect two or more pipes to one (or vice versa) are allowed. Finally, Alice connects a water tap to one of her ends of the pipes, e.g., to pipe 3 and she turns on the tap. Alice and Bob observe which of the two gardens gets sprinkled. It is easy to see that since Alice and Bob only use simple one-to-one connections, there is no "deadlock" possible and the water will indeed eventually come out on one of the two sides. Which side it is obviously depends on the respective local connections.

Now, say that Alice connects her ends of the pipes (and the tap) not in a *fixed* way, but her choice of connections depends on a private bit string $x \in \{0,1\}^n$; for different strings x and x', she may connect her ends of the pipes differently. Similarly, Bob's choice which pipes to connect depends on a private bit string $y \in \{0,1\}^n$. These strategies then specify a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ as follows: $f(x,y)$ is defined to be 0 if, using the connections determined by x and y respectively, the water ends up on Alice's side, and $f(x,y)$ is 1 if the water ends up on Bob's side.

Switching the point of view, we can now take an *arbitrary* Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and ask: How can f be computed in the garden-hose model? How do Alice and Bob have to choose their local connections, and how many water pipes are necessary for computing f in the garden-hose model? We stress that Alice's choice for which pipes to connect may only depend on x but not on y, and vice versa; this is what makes the above questions non-trivial.

In this chapter, we introduce and put forward the notion of *garden-hose complexity.* For a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$, the garden-hose complexity $GH(f)$ of f is defined to be the minimal number s of water pipes needed to compute f in the garden-hose model. It is not too hard to see that $GH(f)$ is well defined (and finite) for any function $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$.

This new complexity notion opens up a large spectrum of natural problems and questions. What is the (asymptotic or exact) garden-hose complexity of natural functions, like equality, inner product *etc.*? How hard is it to compute the garden-hose complexity in general? How is the garden-hose complexity related to other complexity measures? What is the impact of randomness, or entanglement? Some of these questions we answer in this chapter; others remain open.

Lower and upper bounds We show a near-linear $\Omega(n/\log(n))$ lower bound on the garden-hose complexity $GH(f)$ for a natural class of functions $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$. This class of functions includes the mod-2 inner-product function, the equality function, and the majority function. For the former two, this bound is close to tight, in that for these two functions we also show a linear upper bound. In the conference version of these results, we showed a quadratic upper bound for the majority function. In follow-up work Klauck and Podder [KP14] exhibited a protocol which used only $O(n \log^3 n)$ pipes, among several other results on the garden-hose model. In different follow-up work, Margalit and Matsliah improved our upper bound for the equality function with the help of the IBM SAT-Solver [MM12, Mar14] to approximately $1.448n$, and the question of how many water pipes are necessary to compute the equality function in the garden-hose model featured as April 2012's "Ponder This" puzzle on the IBM website[1]. The current best construction was found by Chiu et al. [CSWX14] using approximately $1.359n$ pipes. The *exact* garden-hose complexity of the equality function is still unknown, though; let alone that of other functions.

By using a counting argument, we show the existence of functions with *exponential* garden-hose complexity, but so far, no such function is known explicitly. We also show, again using counting, the existence of a hierarchy – adding more pipes enable the parties to compute strictly more functions.

Connections to other complexity notions We show that every function $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ that is *log-space computable* has polynomial garden-hose complexity. And, vice versa, we show that every function with polynomial garden-hose complexity is, up to local pre-processing, log-space computable. As a consequence, we obtain that the set of functions with polynomial garden-hose complexity is exactly given by the functions that can be computed by arbitrary local pre-processing followed by a log-space computation.

---

[1] `http://ibm.co/I7yvMz`

We also point out a connection to communication complexity by observing that, for any function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, the *one-way communication complexity* of f is a lower bound on $GH(f) \log(GH(f))$.

Randomized and quantum garden-hose complexity We consider the following natural variants of the garden-hose model. In the *randomized* garden-hose model, Alice and Bob additionally share a uniformly random string r, and the water is allowed to come out on the wrong side with small probability ε. Similarly, in the *quantum* garden-hose model, Alice and Bob additionally hold an arbitrary entangled quantum state and their wiring strategies can depend on the outcomes of measuring this state before performing a the garden-hose protocol. Again, the water is allowed to come out on the wrong side with small probability ε. Based on the observed connections of the garden-hose complexity to log-space computation and to one-way communication complexity, we can show that the resulting notion of *randomized* garden-hose complexity $GH_\varepsilon(f)$ is polynomially related to $GH(f)$. For the resulting notion of *quantum* garden-hose complexity $GH^Q_\varepsilon(f)$, we can show a separation (for a partial function) from $GH_\varepsilon(f)$.

Application to quantum position-verification Finally, we show an interesting connection between the garden-hose model and the (in)security of a certain class of *quantum position-verification* schemes. The goal of position-verification is to verify the geographical position *pos* of a prover P by means of sending messages to P and measuring the time it takes P to reply. Position-verification with security against collusion attacks, where different attacking parties collaborate in order to try to fool the verifiers, was shown to be impossible in the classical setting by [CGMO09], and in the quantum setting by [BCF$^+$11], if there is no restriction put upon the attackers. In the quantum setting, this raises the question whether there exist schemes that are secure in case the attackers' quantum capabilities are limited.

Each position-verification scheme $\mathsf{PV}_{\text{qubit}}$ we consider here is specified by a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. This class of schemes was first considered by Kent et al. [KMS11]. These schemes may have the desirable property that the more classical resources the honest users use to faithfully execute the scheme, the more quantum resources the adversary needs in order to break it. It turns out that there is a one-to-one correspondence between the garden-hose model and a certain class of attacks on these schemes, where the attackers teleport a qubit back and forth using a supply of EPR pairs. As an immediate consequence, the (quantum) garden-hose complexity of f gives an upper bound on the number of EPR pairs the attackers need in order to break the scheme $\mathsf{PV}_{\text{qubit}}$. As a corollary, we obtain the following interesting connection between proving the security of quantum protocols and classical complexity theory: If there is an f in P such that there is no way of attacking scheme $\mathsf{PV}_{\text{qubit}}$ using a polynomial number of EPR pairs, then P $\neq$ L. Vice versa, our approach may lead to practical secure quantum position-verification schemes whose security is based on classical complexity-theoretical assumptions such as P is different from L. However, so far it is still unclear whether the (quantum) garden-hose complexity by any means gives a *lower bound* on the number of EPR pairs needed; this remains to be further investigated.

3.2 A scheme for position-verification

The results of this chapter are motivated by the study of a particular quantum protocol for secure position verification, $\mathsf{PV}_{\text{qubit}}$, described in Figure 3.1[2].

In Step 0, the verifiers prepare challenges for the prover. In Step 1, they send the challenges, timed in such a way that they all arrive at the same time at the prover. In Step 2, the prover computes his answers and sends them back to the verifiers. Finally, in Step 3, the verifiers verify the timing and correctness of the answer.

As in [BCF$^+$11], we consider here for simplicity the case where all players live in one dimension, the basic ideas generalize to higher dimensions. In one dimension, we can focus on the case of two verifiers V_0, V_1 and an honest prover P in between them.

We minimize the amount of quantum communication in that only one verifier, say V_0, sends a qubit to the prover, whereas both verifiers send classical n-bit strings $x, y \in \{0,1\}^n$ that arrive at the same time at the prover. We fix a publicly known Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ whose output $f(x, y)$ decides whether the prover has to return the qubit (unchanged) to verifier V_0 (in case $f(x, y) = 0$) or to verifier V_1 (if $f(x, y) = 1$).

The motivation for considering this protocol is the following: As the protocol uses only one qubit which needs to be correctly routed, the honest prover's quantum actions are trivial to perform. His main task is evaluating a classical Boolean function f on classical inputs x and y whose bit size n can be easily scaled up. On the other hand, our results in this section suggest that the adversary's job of successfully attacking the protocol becomes harder and harder for larger input strings x, y. The hope is that for "complicated enough" functions $f(x, y)$, the amount of EPR pairs (ebits) needed to successfully break the security of the protocol $\mathsf{PV}_{\text{qubit}}$ grows (at least) linearly in the bit length n of the classical strings x, y.

If this intuition can be proven to be true, it is a very interesting property of the protocol that we obtain a favorable relation between quantum and classical difficulty of operations in the following sense: if we increase the length of the

---

[2] The protocol is of the generic form described in Section 3.2 of [BCF$^+$11].

0. V_0 randomly chooses two n-bit strings $x, y \in \{0,1\}^n$ and privately sends y to V_1. V_0 prepares an EPR pair $(|0\rangle_V|0\rangle_P + |1\rangle_V|1\rangle_P)/\sqrt{2}$. If $f(x,y) = 0$, V_0 keeps the qubit in register V. Otherwise, V_0 sends the qubit in register V privately to V_1.
1. V_0 sends the qubit in register P to the prover P together with the classical n-bit string x. V_1 sends y so that it arrives at the same time as the information from V_0 at P.
2. P evaluates $f(x,y) \in \{0,1\}$ and routes the qubit to $V_{f(x,y)}$.
3. V_0 and V_1 accept if the qubit arrives in time at the right verifier and the Bell measurement of the received qubit together with the qubit in V yields the correct outcome.

Figure 3.1: Position-verification scheme $\mathsf{PV}_{\text{qubit}}$ using a single qubit and classical n-bit strings.

classical inputs x, y, we require more *classical* computing power of the honest prover, whereas more *quantum* resources (ebits) are required by the adversary to break the protocol. To the best of our knowledge, such a trade-off has never been observed for a quantum-cryptographic protocol.

In order to analyze the security of the protocol $\mathsf{PV}_{\text{qubit}}$, we define the following communication game in which Alice and Bob play the roles of the adversarial attackers of $\mathsf{PV}_{\text{qubit}}$. Alice starts with an unknown qubit $|\phi\rangle$ and a classical n-bit string x while Bob holds the n-bit string y. They also share some quantum state $|\eta\rangle_{AB}$ and both players know the Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. The players are allowed one round of simultaneous classical communication combined with arbitrary local quantum operations. When $f(x,y) = 0$, Alice should be in possession of the state $|\phi\rangle$ at the end of the protocol and on $f(x,y) = 1$, Bob should hold it.

As a simple example consider the case where $f(x,y) = x \oplus y$, the exclusive OR function, with 1-bit inputs x and y. Alice and Bob then have the following way of performing this task perfectly by using a pre-shared quantum state consisting of three EPR pairs (three ebits). Label the first two EPR pairs 0 and 1. Alice teleports $|\phi\rangle$ to Bob using the pair labeled with her input x. This yields measurement result $i \in \{0,1,2,3\}$, while Bob teleports his half of the EPR pair labeled y to Alice using his half of the third EPR pair while obtaining measurement outcome $j \in \{0,1,2,3\}$. In the round of simultaneous communication, both players send the classical measurement results and their inputs x or y to the other player. If $x \oplus y = 1$, i.e. x and y are different bits, Bob can apply the Pauli operator σ_i to his half of the EPR pair labeled $x = y \oplus 1$, correctly recovering $|\phi\rangle$. Similarly, if $x \oplus y = 0$, it is easy to check that Alice can recover the qubit by applying $\sigma_i \sigma_j$ to her half of the third EPR pair.

The garden-hose model is inspired by the following question: What attacks are

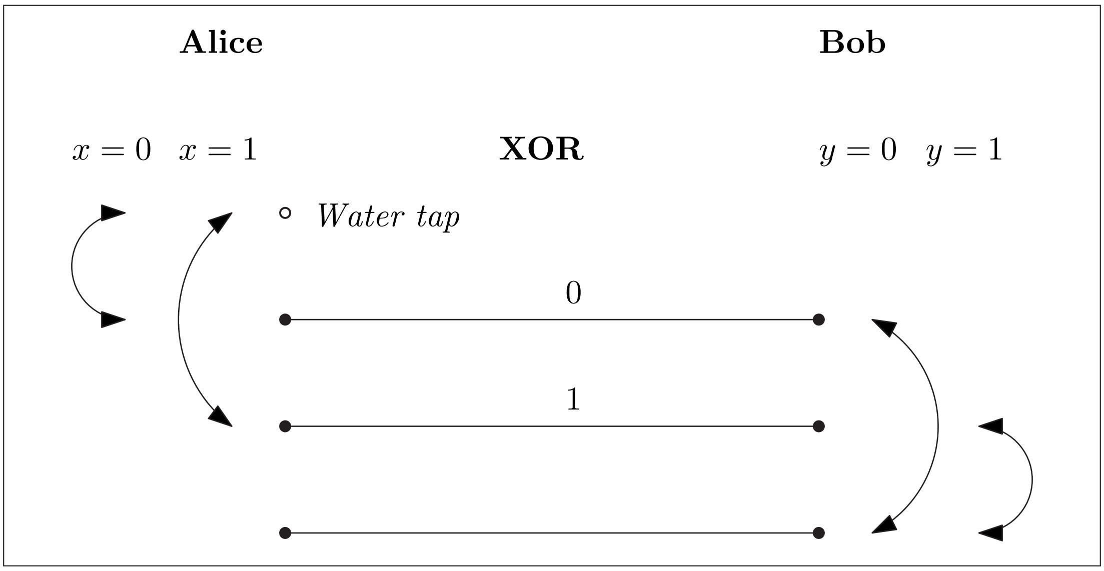

Figure 3.2: Computing the XOR function in the garden-hose model using three water pipes. If Alice's input bit x is 0, she connects the water tap to the first water pipe labeled "0". In case $x = 1$, she connects the tap to the second pipe labeled "1".

possible if Alice and Bob are *constrained* to the types of actions in the example above, i.e., if they are restricted to teleporting the quantum state back and forth depending on their classical inputs?

3.3 The garden-hose model

3.3.1 Definition

Alice and Bob get n-bit input strings x and y, respectively. Their goal is to "compute" an agreed-upon Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ on these inputs, in the following way. Alice and Bob have s water pipes between them, and, depending on their respective classical inputs x and y, they connect (some of) their ends of the pipes with pieces of hose. Additionally, Alice connects a water tap to one of the pipes. They succeed in computing f in the garden-hose model, if the water comes out on Alice's side whenever $f(x, y) = 0$, and the water comes out on Bob's side whenever $f(x, y) = 1$. Note that it does not matter out of which pipe the water flows, only on which side it flows. What makes this task non-trivial is that Alice and Bob must do their "plumbing" based on their local input only, and they are not allowed to communicate. We refer to Figure 3.2 for an illustration of computing the XOR function in the garden-hose model.

We formalize the above description of the garden-hose model, given in terms of pipes and hoses *etc.*, by means of rigorous graph-theoretic terminology. However, we feel that the above terminology captures the notion of the garden-hose model very well, and thus we sometimes use the above "watery" terminology. We start with a balanced bi-partite graph $(A \cup B, E)$ which is 1-regular and where the cardinality of A and B is $|A| = |B| = s$, for an arbitrarily large $s \in \mathbb{N}$. We slightly abuse notation and denote both the vertices in A and in B by the integers $1, \ldots, s$. If we need to distinguish $i \in A$ from $i \in B$, we use the notation i^A and i^B. We may assume that E consists of the edges that connect $i \in A$ with $i \in B$ for every $i \in \{1, \ldots, s\}$, i.e., $E = \left\{\left\{i^A, i^B\right\} : 1 \leq i \leq s\right\}$. These edges in E are the *pipes* in the above terminology. We now extend the graph to $(A_\circ \cup B, E)$ by adding a vertex 0 to A, resulting in $A_\circ = A \cup \{0\}$. This vertex corresponds to the *water tap*, which Alice can connect to one of the pipes. Given a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, consider two functions $E_{A_\circ}$ and E_B; both take as input a string in $\{0,1\}^n$ and output a set of edges (without self loops). For any $x, y \in \{0,1\}^n$, $E_{A_\circ}(x)$ is a set of edges on the vertices $A_\circ$ and $E_B(x)$ is a set of edges on the vertices B, so that the resulting graphs $(A_\circ, E_{A_\circ}(x))$ and $(B, E_B(y))$ have maximum degree at most 1. $E_{A_\circ}(x)$ consists of the *connections* among the pipes (and the tap) on Alice's side (on input x), and correspondingly for $E_B(y)$. For any $x, y \in \{0,1\}^n$, we define the graph $G(x,y) = (A_\circ \cup B, E \cup E_{A_\circ}(x) \cup E_B(y))$ by adding the edges $E_{A_\circ}(x)$ and $E_B(y)$ to E. $G(x,y)$ consists of the pipes with the connections added by Alice and Bob. Note that the vertex $0 \in A_\circ$ has degree at most 1, and the graph $G(x,y)$ has maximum degree at most two 2; it follows that the maximal path $\pi(x,y)$ that starts at the vertex $0 \in A_\circ$ is uniquely determined. $\pi(x,y)$ represents the flow of the water, and the endpoint of $\pi(x,y)$ determines whether the water comes out on Alice's or on Bob's side (depending on whether the final vertex is in $A_\circ$ or in B).

3.3.1. Definition. A *garden-hose protocol* is given by a graph function $G : (x,y) \mapsto G(x,y)$ as described above. The number of pipes s is called the *size* of G, and is denoted as $s(G)$. A garden-hose protocol G is said to *compute* a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ if the endpoint of the maximal path $\pi(x,y)$ starting at 0 is in $A_\circ$ whenever $f(x,y) = 0$ and in B whenever $f(x,y) = 1$.

3.3.2. Definition. The deterministic *garden-hose complexity* of a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is the size $s(G)$ of the smallest garden-hose protocol G that computes f. We denote it by $GH(f)$.

Relation between the garden-hose model and attack on $\mathsf{PV}_{\text{qubit}}$ If Alice and Bob are *constrained* to the types of actions in the example above, i.e., if they are restricted to teleporting the quantum state back and forth depending on their classical inputs, there is a one-to-one correspondence between attacking the position-verification scheme $\mathsf{PV}_{\text{qubit}}$ and computing the function f in the garden-hose model. The quantum strategy for attacking $\mathsf{PV}_{\text{qubit}}$ in the example above exactly corresponds to the strategy depicted in Figure 3.2 for computing the XOR-function in the garden-hose model.

More generally, we can translate any strategy of Alice and Bob in the garden-hose model to a perfect quantum attack of $\mathsf{PV}_{\text{qubit}}$ by using one EPR pair per pipe and performing Bell measurements where the players connect the pipes.

Our hope is that also the converse is true: if many pipes are required to compute f (say we need super-polynomially many), then the number of EPR pairs needed for Alice and Bob to successfully break $\mathsf{PV}_{\text{qubit}}$ with probability close to 1 by means of an *arbitrary* attack (not restricted to Bell measurements on EPR pairs) should also be super-polynomial.

The examples of (partial) functions from Theorem 3.5.2 show that the classical garden-hose complexity $GH(f)$ does not capture the amount of EPR pairs required to attack $\mathsf{PV}_{\text{qubit}}$. It is conceivable that one can show that arbitrary attacks can be cast in the quantum garden-hose model and hence, the quantum garden-hose complexity $GH^Q_\varepsilon(f)$ (or a variant of it[3]) correctly captures the amount of EPR pairs required to attack $\mathsf{PV}_{\text{qubit}}$. We leave this question as an interesting problem for future research.

We stress that for this application, any polynomial lower bound on the number of required EPR pairs is already interesting.

3.3.2 Upper and lower bounds

In this section, we present upper and lower bounds on the number of pipes required to compute some particular (classes of) functions in the garden-hose model. We first give a simple upper bound on $GH(f)$ which is implicitly proven in the attack on Scheme II in [KMS11].

3.3.3. Proposition. *For every Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, the garden-hose complexity $GH(f)$ is at most $2^n + 1$.*

Proof. We identify $\{0,1\}^n$ with $\{1,\dots,2^n\}$ in the natural way. For $s = 2^n + 1$ and the resulting bipartite graph $(A_\circ \cup B, E)$, we can define $E_{A_\circ}$ and E_B as follows. $E_{A_\circ}(x)$ is set to $\{(0,x)\}$, meaning that Alice connects the tap with the pipe labeled by her input x. To define E_B, group the set $Z(y) = \{a \in \{0,1\}^n : f(a,y) = 0\}$ arbitrarily into disjoint pairs $\{a_1,a_2\} \cup \{a_3,a_4\} \cup \ldots \cup \{a_{\ell-1},a_\ell\}$ and set $E_B(y) = \{\{a_1,a_2\},\{a_3,a_4\},\dots,\{a_{\ell-1},a_\ell\}\}$. If $\ell = |Z(y)|$ is odd so that the decomposition into pairs results in a left-over $\{a_\ell\}$, then a_ℓ is connected with the "reserve" pipe labeled by $2^n + 1$.

By construction, if $x \in Z(y)$ then $x = a_i$ for some i, and thus pipe $x = a_i$ is connected on Bob's side with pipe a_{i-1} or a_{i+1}, depending on the parity of i, or with the "reserve" pipe, and thus $\pi(x,y)$ is of the form $\pi(x,y) = (0, x^A, x^B, v^B, v^A)$, ending in $A_\circ$. On the other hand, if $x \notin Z(y)$, then pipe x is not connected on Bob's side, and thus $\pi(x, y) = (0, x^A, x^B)$, ending in B. This proves the claim. □

---

[3] In addition to the number of pipes, one might have to account for the size of the entangled state as well.

We notice that we can extend this proof to show that the garden-hose complexity $GH(f)$ is at most $2^{D(f)+1} - 1$, where $D(f)$ is the deterministic communication complexity of f.

3.3.4. PROPOSITION. *The garden-hose complexity $GH(f)$ of any function f is at most $2^{D(f)+1} - 1$, where $D(f)$ is the deterministic communication complexity of f.*

Proof. Consider a protocol where Alice and Bob alternate in sending one bit. The pipes between Alice and Bob are labeled with all possible non-empty strings of length up to $D(f)$, with one extra reserve pipe.

Let $A_v(x)$ be the bit Alice sends after seeing transcript $v \in \{0,1\}^*$ given input x and let $B_v(x)$ be the bit Bob sends after a transcript v on input y. (Since Alice and Bob alternate, Alice sends a bit on even length transcripts, while Bob sends when the transcript has odd length.) Alice connects the tap to 0 or 1 depending on the first sent bit. Then, Alice makes connections

$$\{\{v, vA_v(x)\} \mid v \in \{0,1\}^* \text{with } |v| \text{ even and } 1 \le |v| \le D(f)\}\,.$$

Here $vA_v(x)$ is the concatenation of v and $A_v(x)$. Bob's connections are given by the set

$$\{\{v, vB_v(x)\} \mid v \in \{0,1\}^* \text{with } |v| \text{ odd and } 1 \le |v| \le D(f)\}\,.$$

Now, for all transcripts of length $D(f)$, Alice knows the function outcome. (Assume $D(f)$ is even for simplicity.) For those $2^{D(f)}$ pipes she can route the water to the correct side by connecting similar outcomes, as in the proof of Proposition 3.3.3, using one extra reserve pipe. This brings the total used pipes to $1 + \sum_{i=1}^{D(f)} 2^i = 2^{D(f)+1} - 1$. The correctness can be verified by comparing the path of the water to the communication protocol: the label of the pipe the water is in, when following it through the pipes for r "steps", is exactly the same as the transcript of the communication protocol when executing it for r rounds. □

3.3.5. DEFINITION. We call a function f *injective for Alice*, if for every two different inputs x and x' there exists y such that $f(x, y) \ne f(x', y)$. We define *injective for Bob* in an analogous way: for every $y \ne y'$, there exists x such that $f(x, y) \ne f(x, y')$ holds.

3.3.6. PROPOSITION. *If f is injective for Bob or f is injective for Alice, then*

$$GH(f) \log(GH(f)) \ge n\,.$$

Proof. We give the proof when f is injective for Bob. The proof for the case where f is injective for Alice is the same. Consider a garden-hose protocol G that computes f. Let s be its size $s(G)$. Since, on Bob's side, every pipe is connected to at most one other pipe, there are at most $s^s = 2^{s\log(s)}$ possible choices for $E_B(y)$, i.e., the set of connections on Bob's side. Thus, if $2^{s\log(s)} < 2^n$, it follows from the pigeonhole principle that there must exist y and y' in $\{0,1\}^n$ for which $E_B(y) = E_B(y')$, and thus for which $G(x,y) = G(x,y')$ for all $x \in \{0,1\}^n$. But this cannot be since G computes f and $f(x,y) \neq f(x,y')$ for some x due to the injectivity for Bob. Thus, $2^{s\log(s)} \geq 2^n$ which implies the claim. □

We can use this result to obtain an almost linear lower bound for several functions that are often studied in communication complexity settings such as:

- Bitwise inner product: $\mathrm{IP}(x,y) = \sum_i x_i y_i \pmod 2$
- Equality: $\mathrm{EQ}(x,y) = 1$ if and only if $x = y$
- Majority: $\mathrm{MAJ}(x,y) = 1$ if and only if $\sum_i x_i y_i \geq \lceil \frac{n}{2} \rceil$

The first two of these functions are injective for both Alice and Bob, while majority is injective for inputs of Hamming weight at least $n/2$, giving us the following corollary.

3.3.7. Corollary. *The functions bitwise inner product, equality and majority have garden-hose complexity in* $\Omega(\frac{n}{\log(n)})$.

By considering the water pipes that actually get wet, one can show a lower bound of n pipes for equality [Pie11]. On the other hand, we can show upper bounds that are linear for the bitwise inner product and equality, and quadratic in case of majority. For illustration, we will present explicit constructions for the equality and inner product function. A simple $O(n^2)$ construction for the majority function can be found in the author's Master's thesis [Spe11], this was later improved to almost-linear by Klauck and Podder [KP14].

3.3.8. Proposition. *In the garden-hose model, the equality function*[4] *can be computed with* $3n+1$ *pipes and the bitwise inner product can be computed with* $4n+1$ *pipes.*

3.3.3 Equality

For a graphical depiction of the protocol, see Figure 3.3. As initialization, Alice first connects the source to pipe R_0, effectively letting Bob start with the water.

---

[4] Also see later follow-up results by different authors [Mar14, CSWX14], that improve on these bounds.

We repeat the same pattern, for every i from 1 to n. If $y = 0$, Bob connects pipe R_{i-1} to pipe Q_i^0, and on $y = 1$, Bob connects pipe R_{i-1} to pipe Q_i^1. On the other side, Alice connects R_i to Q_i^0 if $x = 0$ and she connects R_i to Q_i^1 instead, if $x = 1$.

If x and y are different on bit j, then $Q_j^{y_j}$ stays unconnected, so the water will flow out on Alice's side, right there. If x and y are equal this situation will never happen, so the water will exit at R_n, on Bob's side. The strategy uses $3n + 1$ pipes, so we have shown that

$$GH(\mathrm{EQ}) \leq 3n + 1.$$

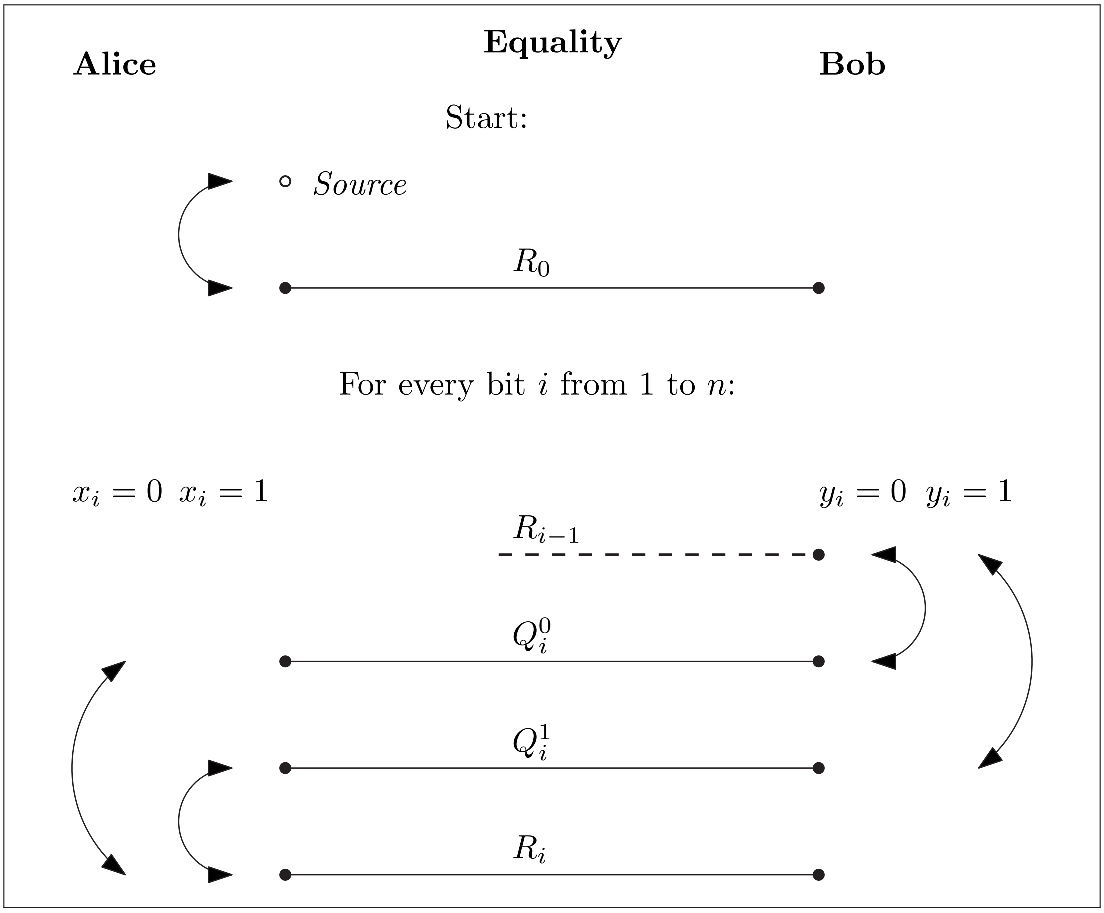
Equality
Alice
Bob
Start:
Source
R_0
For every bit i from 1 to n:
$x_i = 0$ $x_i = 1$
$y_i = 0$ $y_i = 1$
R_{i-1}
Q_i^0
Q_i^1
R_i

Figure 3.3: Garden-hose protocol for the equality function.

3.3.4 Inner product

The protocol for inner product is drawn in Figure 3.4. Recall that the inner-product function is defined as $\mathrm{IP}(x, y) = \sum_i x_i y_i \pmod 2$. Consider the following simple algorithm to calculate the bitwise inner product: Initialize a one-bit result register with the value 0, let i step from 1 to n, and flip the register bit whenever the AND of x_i and y_i equals 1. The garden-hose protocol follows a strategy inspired by this idea.

To start, Alice connects the source to Q_k^0, with k the first index for which $x_k = 1$. For every i from 1 to n, there are four pipes. If $y_i = 0$, Bob connects Q_i^0 to R_i^0 and Q_i^1 to R_i^1. If $y = 1$, Bob instead connects Q_i^0 to R_i^1 and Q_i^1 to R_i^0.

Alice does not make any new connections if $x_i = 0$, and if $x_i = 1$ she connects R_i^0 and R_i^1 to R_k^0 and R_k^1 respectively, with k the next index for which $x_k = 1$. If x_i is the last bit of x equal to 1, Alice does nothing with R_i^0 and connects R_i^1 to the pipe labeled *End*.

To see why this construction works, we can compare it to the simple algorithm described earlier. The water flowing through R_i^b corresponds to the result register having value b after step i of the algorithm, and the water changes from the top to bottom pipe, or vice versa, when $x_i = y_i = 1$. At the last index k for which $x_k = 1$, the water flows to Alice through the pipe corresponding to the final function value. Alice leaves R_k^0 unconnected, so the water exits at Alice's side if $\mathrm{IP}(x, y) = 0$. She connects R_k^1 to the pipe *End*, which is unconnected on Bob's side, making the water exit at Bob's side if $\mathrm{IP}(x, y) = 1$.

The strategy uses $4n + 1$ pipes, letting us upper bound the garden-hose complexity with

$$GH(\mathrm{IP}) \leq 4n + 1.$$

3.3.5 Lower bounds

3.3.9. Proposition. *There exist functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ for which $GH(f)$ is exponential.*

Proof. The existence of functions with an exponential garden-hose complexity can be shown by a simple counting argument. There are $2^{2^{2n}}$ different functions $f(x, y)$. For a given size $s = s(G)$ of G, for every $x \in \{0,1\}^n$, there are at most $(s+1)^{s+1}$ ways to choose the connections $E_{A_\circ}(x)$ on Alice's side, and thus there are at most $((s+1)^{s+1})^{2^n} = 2^{2^n(s+1)\log(s+1)}$ ways to choose the function $E_{A_\circ}$. Similarly for E_B, there are at most $2^{2^n s \log(s)}$ ways to choose E_B. Thus, there are at most $2^{2 \cdot 2^n(s+1)\log(s+1)}$ ways to choose G of size s. Clearly, in order for every function f to have a G of size s that computes it, we need that $2 \cdot 2^n(s+1)\log(s+1) \geq 2^{2n}$, and thus that $(s + 1)\log(s + 1) \geq 2^{n-1}$, which means that s must be exponential. □

We can extend this result, with the same choice as functions as [BCP+13], to show a hierarchy of functions that can be computed depending on the number of pipes:

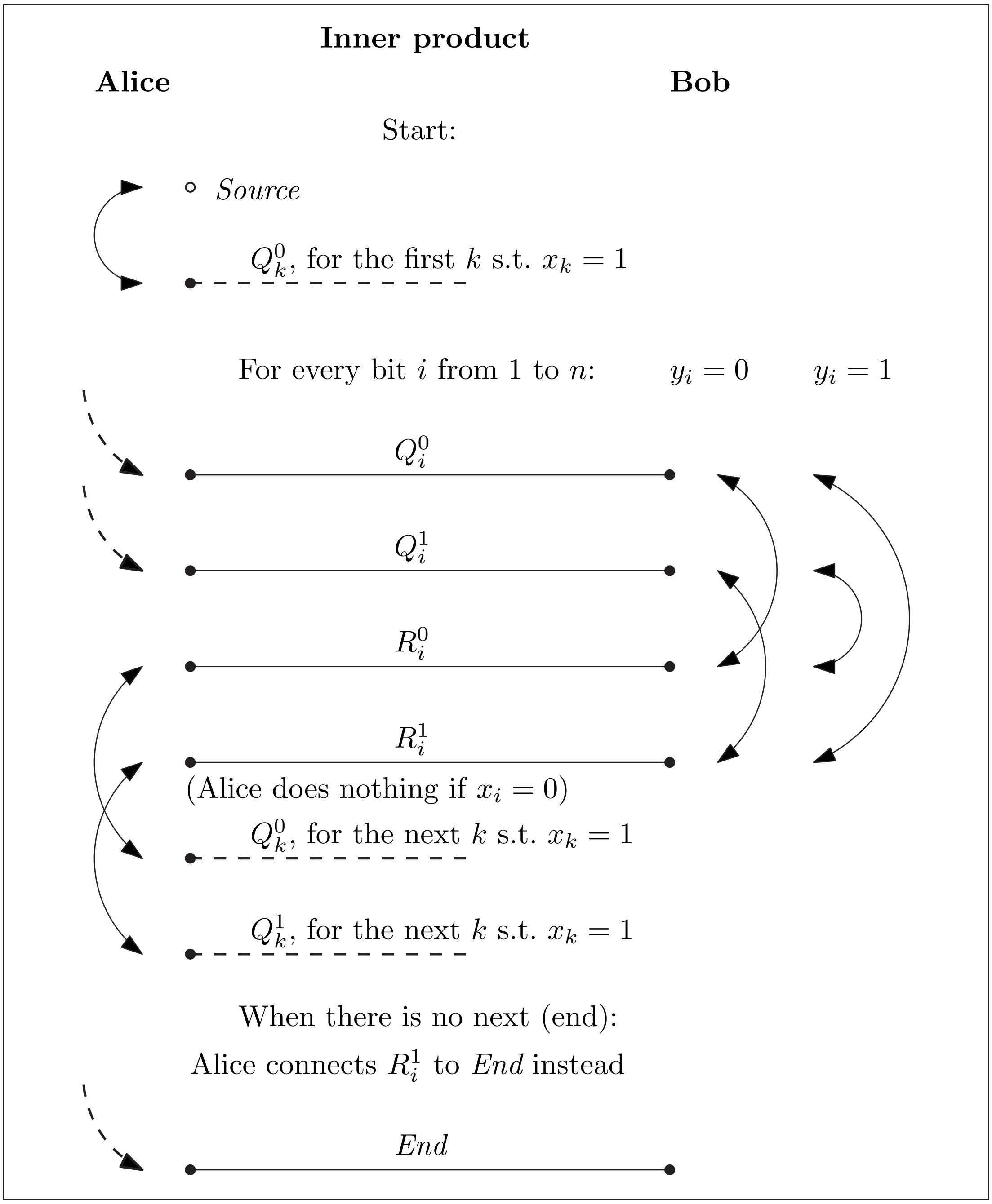
Inner product
Alice
Bob
Start:
Source
Q_k^0, for the first k s.t. $x_k = 1$
For every bit i from 1 to n:
$y_i = 0$
$y_i = 1$
Q_i^0
Q_i^1
R_i^0
R_i^1
(Alice does nothing if $x_i = 0$)
Q_k^0, for the next k s.t. $x_k = 1$
Q_k^1, for the next k s.t. $x_k = 1$
When there is no next (end):
Alice connects R_i^1 to *End* instead
End

Figure 3.4: Garden-hose protocol for the inner-product function.

3.3.10. Proposition. *Given any natural numbers s and s' such that $s' > 2(s+1)\log(s+1)+1$ and $s' \leq 2^n+1$, there exist functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that can be computed by s' pipes but can not be computed by s pipes.*

Proof. Let $\#f(s)$ denote the number of $2n$-bit functions for which there exists a garden-hose protocol with s pipes. From the proof of Proposition 3.3.9, we know that $\#f(s) \leq 2^{2(s+1)\log(s+1)2^n}$.

To show a lower bound on the number of functions $\#f(s')$, note that the construction of Proposition 3.3.3 only depends on the 2^n different possibilities of Alice's input x, and has no dependence on the length of Bob's input y.

We will count all functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that are only non-trivial for the first $s'-1$ possible values of x, and are 0 otherwise. There are $2^{(s'-1)2^n}$ such n-bit functions: since x can take $(s'-1)$ different values, and y can take all 2^n possibilities, we consider $(s'-1)2^n$ input pairs, and for each pair the function can have output value either 0 or 1. Each of these functions has a simple garden-hose protocol using s' pipes, by the construction of Proposition 3.3.3.

There exists a function which has a garden-hose protocol with s' pipes but not with s pipes whenever $\#f(s) < \#f(s')$, which, filling in the earlier upper and lower bounds, is implied by

$$2^{2(s+1)\log(s+1)2^n} < 2^{(s'-1)2^n},$$

that is, when $2(s+1)\log(s+1) < s'-1$. □

A natural class of functions we might use in cryptographic settings are those of the form

$$f(x,y) = g(x \oplus y),$$

where we start with some n-bit function and turn it into a $2n$-bit function by taking the XOR of the inputs x and y. There are specific functions of this form that we can lower bound using the observation of Proposition 3.3.12, for example functions which depend on the Hamming distance of these strings [AGSU15], but one could also wonder whether there exist functions of this type with exponential garden-hose complexity. It is possible to show (again via a simple observation and a counting argument) that there exists a promise problem for which it is the case [Sze12].

3.3.11. Proposition. *There exist (partial) functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ such that $f(x,y) = g(x \oplus y)$ for some $g : \{0,1\}^n \to \{0,1\}$ for which $GH(f)$ is exponential.*

Proof. Consider functions with a promise that $x = (x', 0^{n/2})$ and $y = (0^{n/2}, y')$. Say Alice and Bob share s pipes between them. There are 2^{2^n} functions $f = g(x \oplus y) = g(x', y')$ of this form, one for any function g, but with this promise Alice and Bob have only $2^{s \log s 2^{n/2}}$ different strategies each. Then we need that $2^{2^n} \leq (2^{s \log s 2^{n/2}})^2$ and therefore s needs to be exponential. □

In general, garden-hose protocols can be transformed into (one-way) communication protocols by Alice sending her connections $E_{A_\circ}(x)$ to receiver Bob, which will require at most $GH(f)\log(GH(f))$ bits of communication. Bob can then locally compute the function by combining Alice's message with $E_B(y)$ and checking where the water exits.[5] We summarize this observation in the following proposition.

3.3.12. PROPOSITION. *Let $D^1(f)$ denote the deterministic one-way communication complexity of f. Then,*

$$D^1(f) \leq GH(f)\log(GH(f)) .$$

As a consequence, lower bounds on the communication complexity carry over to the garden-hose complexity (up to logarithmic factors). Notice that this technique will never give lower bounds that are better than linear, as problems in communication complexity can always be solved by sending the entire input to the other party. It remains an interesting open problem to show super-linear lower bounds in the garden-hose model . The majority function was named as a likely candidate for needing a super-linear number of pipes in the original paper that presented the garden-hose model [BFSS13], but recently Klauck and Podder showed that an $O(n \log^3 n)$ garden-hose protocol for the function exists [KP14].

3.3.6 Polynomial garden-hose complexity and log-space computations

A family of Boolean functions $\{f_n\}_{n\in\mathbb{N}}$ is *log-space computable* if there exists a deterministic Turing machine M and a constant c, such that for any n-bit input x, M outputs the correct output bit $f_n(x)$, and at most $c \cdot \log n$ locations of M's work tapes are ever visited by M's head during computation.

3.3.13. DEFINITION. We define the complexity class $\mathrm{L}_{(2)}$, called *logarithmic space with local pre-processing*, to be the class of Boolean functions $f(x, y)$ for which there exists a Turing machine M and two *arbitrary* functions $\alpha(x), \beta(y)$, such that[6] $M(\alpha(x), \beta(y)) = f(x, y)$ and $M(\alpha(x), \beta(y))$ runs in space logarithmic in the size of the original inputs $|x| + |y|$.

This definition can be extended in a natural way by considering Turing machines and circuits corresponding to various complexity classes, and by varying the number of players. For example, a construction as in Proposition 3.3.3 and a similar reasoning as in Proposition 3.3.17 below can be used to show that every Boolean function is contained in $\mathrm{PSPACE}_{(2)}$. As main result of this section, we show that our newly defined class $\mathrm{L}_{(2)}$ is equivalent to functions with polynomial garden-hose complexity. We leave it for future research to study intermediate classes such as $\mathrm{AC}^0_{(2)}$ which are related to the polynomial hierarchy of communication complexity [BFS86].

---

[5] In fact, garden-hose protocols can even be transformed into communication protocols in the more restrictive *simultaneous-message-passage* model, or SMP, where Alice and Bob send simultaneous messages consisting of their connections $E_{A_\circ}(x)$ and $E_B(y)$ to the referee who then computes the function. The according statements of Propositions 3.3.12, 3.4.5 and 3.5.1 can be derived analogously.

[6] For simplicity of notation, we give two arguments to the Turing machine whose concatenation $(\alpha(x), \beta(y))$ is interpreted as the input.

3.3.14. Theorem. *The set of functions f with polynomial garden-hose complexity $GH(f)$ is equal to* $\mathrm{L}_{(2)}$.

The two directions of the theorem follow from Theorem 3.3.15 and Proposition 3.3.17.

3.3.15. Theorem. *If $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is log-space computable, then $GH(f)$ is polynomial in n.*

Our proof explicitly encodes the configuration graph of a Turing machine, and then lets the water follow the computation. A different feasible proof strategy, which we do not follow here, would be to first use the completeness of multiplication of permutations, elements of S_n, for L, under suitable reductions [CM87, IL95], because of the close relation between the garden-hose model and permutations.

Proof. Let M be a deterministic Turing machine deciding $f(x, y) = 0$. We assume that M's read-only input tape is of length $2n$ and contains x on positions 1 to n and y on positions $n + 1$ to $2n$. By assumption M uses logarithmic space on its work tapes.

In this proof, a *configuration* of M is the location of its tape heads, the state of the Turing machine and the content of its work tapes, excluding the content of the read-only input tape. This is a slightly different definition than usual, where the content of the input tape is also part of a configuration. When using the normal definition (which includes the content of all tapes), we will use the term *total configuration.* Any configuration of M can be described using a logarithmic number of bits, because M uses logarithmic space.

A Turing machine is called *deterministic*, if every total configuration has a unique next configuration. A Turing machine is called *reversible*, if in addition to being deterministic, every total configuration also has a unique predecessor. It was shown by Lange, McKenzie and Tapp that any $S(n)$ space-bounded deterministic Turing machine can be simulated by a reversible Turing machine in space $O(S(n))$ [LMT97]. This means that without loss of generality, we can assume M to be a reversible Turing machine, which is crucial for our construction.

Any state of the Turing machine is either *moving*, meaning that the tape head moves, or *stationary.* The proof of [LMT97] also immediately gives us the convenient property that all moving states are *oblivious*, meaning that the direction that the tape head moves is only determined by the state of the Turing machine, and not the contents under the tape head.

Alice's and Bob's strategies in the garden-hose model are as follows. They list all (moving) configurations where the head of the input tape is on position n and about to move to position $n+1$. Let us call the set of these configurations $C_{A\to B}$. Let $C_{B\to A}$ be the analogous set of configurations where the input tape head is on position $n+1$ and is about to move to position n. Because M is oblivious on the moving configurations, these sets depend only on the function f, but not on the input pair (x, y). The number of elements of $C_{A\to B}$ and $C_{B\to A}$ is at most polynomial, being exponential in the description length of the configurations. Now, for every element in $C_{A\to B}$ and $C_{B\to A}$, the players label a pipe with this configuration. Also label $|C_{B\to A}|$ pipes ACCEPT and $|C_{A\to B}|$ of them REJECT. These steps determine the number of pipes needed, Alice and Bob can do this labeling beforehand.

For every configuration in $C_{B\to A}$, with corresponding pipe p, Alice runs the Turing machine starting from that configuration until it either accepts, rejects, or until a configuration from $C_{A\to B}$ is encountered, i.e., until the head is about to move to position $n+1$. If the Turing machine accepts, Alice connects p to the first free pipe labeled ACCEPT. On a reject, she leaves p unconnected. If the tape head of the input tape is about to move to position $n+1$, she connects p to the pipe from $C_{A\to B}$ corresponding to the configuration of the Turing machine she is simulating. By her knowledge of x, Alice knows the content of the input tape on positions 1 to n, but not the other half. Alice also runs M from the starting configuration, connecting the water tap to a target pipe with a configuration from $C_{A\to B}$ depending on the reached configuration.

Bob connects the pipes labeled by $C_{A\to B}$ in an analogous way: He runs the Turing machine starting with the configuration with which the pipe is labeled until it halts or the position of the input tape head is about to move to n. On accepting, the pipe is left unconnected and if the Turing machine rejects, the pipe is connected to one of the pipes labeled REJECT. Otherwise, the pipe is connected to the one labeled with the configuration in $C_{B\to A}$, the configuration the Turing machine is in when the head on the input tape is moving to n.

In the garden-hose model, only one-to-one connections of pipes are allowed. Therefore, to check that the described strategy is a valid one, the simulations of two different configurations from $C_{B\to A}$ should never reach the same configuration in $C_{A\to B}$ and vice-versa. This is guaranteed by the reversibility of M as follows. Consider Alice simulating M starting from different configurations $c, c' \in C_{B\to A}$. We have to check that their simulation can not end at the same $d \in C_{A\to B}$, because Alice can not connect both pipes labeled c and c' to the same d. Because M is reversible, we can in principle also simulate M backwards in time starting from a certain configuration. In particular, Alice can simulate M backwards starting with configuration d, until the input tape head position reaches $n+1$. The configuration of M at that time can not simultaneously be c and c', so there will never be two different pipes trying to connect to the pipe labeled d.

It remains to show that, after the players link up their pipes as described, the water comes out on Alice's side if M rejects on input (x, y), and that otherwise the water exits at Bob's. We can verify the correctness of the described strategy by comparing the flow of the water directly to the execution of M. Every pipe the water flows through corresponds to a configuration of M when it runs starting from the initial state. So the side on which the water finally exits also corresponds to whether M accepts or rejects. □

In the garden-hose model, we allow Alice and Bob to locally pre-process their inputs before computing their wiring. Therefore, it immediately follows from Theorem 3.3.15 that any function f in $\mathrm{L}_{(2)}$ has polynomial garden-hose complexity, proving one direction of Theorem 3.3.14.

We saw in Proposition 3.3.9 that there exist functions with large garden-hose complexity. However, a negative implication of Theorem 3.3.15 is that proving the existence of a *polynomial-time computable* function f with exponential garden-hose complexity is at least as hard as separating L from P, a long-standing open problem in complexity theory.

3.3.16. Corollary. *If there exists a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ in* P *that has super-polynomial garden-hose complexity, then* P $\neq$ L*.*

It remains to prove the other inclusion of Theorem 3.3.14.

3.3.17. Proposition. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a Boolean function. If $GH(f)$ is polynomial (in n), then f is in* $\mathrm{L}_{(2)}$*.*

Proof. Let G be the garden-hose protocol that achieves $s(G) = GH(f)$. We write s for $s(G)$, the number of pipes, and we let $E_{A_\circ}$ and E_B be the underlying edge-picking functions, which on input x and y, respectively, output the connections that Alice and Bob apply to the pipes. Note that by assumption, s is polynomial. Furthermore, by the restrictions on $E_{A_\circ}$ and E_B, on any input, they consist of at most $(s + 1)/2$ connections.

We need to show that f is of the form $f(x, y) = g(\alpha(x), \beta(y))$, where α and β are arbitrary functions $\{0,1\}^n \to \{0,1\}^m$, $g : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ is log-space computable, and m is polynomial in n. We define α and β as follows. For any $x, y \in \{0,1\}^n$, $\alpha(x)$ is simply a natural encoding of $E_{A_\circ}(x)$ into $\{0,1\}^m$, and $\beta(y)$ is a natural encoding of $E_B(y)$ into $\{0,1\}^m$. In the hose-terminology we say that $\alpha(x)$ is a binary encoding of the connections of Alice, and $\beta(y)$ is a binary encoding of the connections of Bob. Obviously, these encodings can be done with m of polynomial size. Given these encodings, finding the endpoint of the maximum path $\pi(x, y)$ starting in 0 can be done with logarithmic space: at any point during the computation, the Turing machine only needs to maintain a pointer to the position of the water and a binary flag to remember on which side of the input tape the head is. Thus, the function g that computes $g(\alpha(x), \beta(y)) = f(x, y)$ is log-space computable in m and thus also in n. □

We also note the connection between the notion of computation with local preprocessing, the garden-hose model, and the recent work on *space-bounded communication complexity*, first defined by Brody et al. [BCP$^+$13]. This connection was made more explicit in the thesis of Song [Son14], as a result of discussions between Song, Buhrman and Speelman in January 2014 at CWI in Amsterdam. In particular, it was shown that the class of functions with polynomial garden-hose complexity, directly corresponds to the functions that have multi-round two-way space-bounded communication protocols that use logarithmic space.

3.4 Randomized garden-hose complexity

It is natural to study the setting where Alice and Bob share a common random string and are allowed to err with some probability ε. More formally, we let the players' local strategies $E_{A_\circ}(x,r)$ and $E_B(y,r)$ depend on the shared randomness r and write $G_r(x,y) = f(x,y)$ if the resulting garden-hose protocol $G_r(x,y)$ computes $f(x,y)$.

3.4.1. Definition. Let r be the shared random string. The *randomized garden-hose complexity* of a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is the size $s(G_r)$ of the smallest garden-hose protocol G_r such that $\forall x, y :\ \Pr_r[G_r(x,y) = f(x,y)] \geq 1 - \varepsilon$. We denote this minimal size by $GH_\varepsilon(f)$.

By repeating the protocol a polynomial number of times the error probability can be made exponentially small.

3.4.2. Proposition. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function such that $GH_\varepsilon(f)$ is polynomial in n, with error $\varepsilon \leq \frac{1}{2} - n^{-c}$ for a constant $c > 0$. For every constant $d > 0$ there exists a polynomial $q(\cdot)$ such that $GH_{2^{-n^d}}(f) \leq q\Big(GH_\varepsilon(f)\Big)$.*

Proof. The new protocol $G'_r(x,y)$ takes the majority of $k = 8n^{2c+d}$ outcomes of $G_{r_i}(x,y)$ where $r_1, \ldots, r_k$ are k independent and uniform samples of the random string. We have to establish (i) that taking the majority of k instances of the original protocol indeed gives the correct outcome with probability at least $1-2^{-n^d}$ and (ii) that $G'_r(x,y)$ requires only polynomial pipes.

(i) Let X_i be the random variable that equals 1 when $G_{r_i}(x,y) = f(x,y)$ and 0 otherwise. Note that the X_i are independent and identically distributed random variables with expectation $E[X_i] \geq 1-\varepsilon =: p$. Whenever $\sum_{i=1}^k X_i \geq \frac{k}{2}$ the protocol gives the correct outcome. Use the Chernoff bound to get

$$\Pr\left[\sum_{i=1}^{k} X_i < (1-\zeta)pk\right] \leq e^{-\frac{\zeta^2}{2}pk}$$

for any small ζ. Picking $\zeta = n^{-c}$, so that $(1-\zeta)pk$ is still greater than $\frac{k}{2}$, and filling in k, we can upper bound the probability of failure by

$$e^{-\frac{8n^{2c+d}}{2n^{2c}}p} \leq 2^{-n^d}$$

(ii) In Theorem 3.3.15 we show that any log-space computable function can be simulated by a polynomial-sized garden-hose strategy. Thus, if checking the majority of k garden-hose strategies can be done in logarithmic space (after local pre-computations by Alice and Bob), then $G'_r(x, y)$ can be computed using a polynomial number of pipes.

Let $A_i = E_{A_\circ}(x, r_i)$ be the local wiring of Alice for strategy G on input x with randomness r_i, and let $B_i = E_B(y, r_i)$. Alice locally generates $(A_1, \ldots, A_k)$ and Bob locally generates $(B_1, \ldots, B_k)$. In the proof of Proposition 3.3.17 it was shown that simulating the outcome of a single garden-hose strategy (A_i, B_i) can be done in logarithmic space. Here we follow the same construction, but instead of getting the outcome of a single strategy we simulate all k strategies. This can still be done in logarithmic space, since we can re-use the memory needed to simulate each of the k strategies. To find the majority, we need to add a counter to keep track of the simulation outcomes, using only an extra $\log k$ bits of space.

□

Using this result, any randomized strategy can be turned into a deterministic strategy with only a polynomial overhead in the number of pipes.

3.4.3. Proposition. *Let* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *be a function such that* $GH_\varepsilon(f)$ *is polynomial in* n *and* $\varepsilon \leq \frac{1}{2} - n^c$ *for a constant* $c > 0$. *Then there exists a polynomial* $q(\cdot)$ *such that* $GH(f) \leq q\Big(GH_\varepsilon(f)\Big)$.

Proof sketch. By Proposition 3.4.2 there exists a randomized garden-hose protocol $G_r(x, y)$ of size $q(GH_\varepsilon(f))$ with error probability at most 2^{-2n-1}. The probability for a random string r to be wrong for all inputs is at most $2^{2n} \cdot 2^{-2n-1} < 1$, by the union bound. In particular, there exists a string $\hat{r}$ which works for every input (x, y). □

Using this Proposition 3.4.3, we conclude that the lower bound from Proposition 3.3.9 carries over to the randomized setting.

3.4.4. Corollary. *There exist functions* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *for which* $GH_\varepsilon(f)$ *is exponential.*

With the same reasoning as in Proposition 3.3.12, we get that lower bounds on the randomized one-way communication complexity with public shared randomness carry over to the randomized garden-hose complexity (up to a logarithmic factor).

3.4.5. Proposition. *Let* $R_{\varepsilon}^{1,\mathrm{pub}}(f)$ *denote the minimum communication cost of a one-way-communication protocol which computes* f *with an error* ε *using public shared randomness. Then,* $R_{\varepsilon}^{1,\mathrm{pub}}(f) \leq GH_{\varepsilon}(f)\log(GH_{\varepsilon}(f))$.

For instance, the linear lower bound $R_{\varepsilon}^{pub}(IP) \in \Omega(n)$ from [CG88] for the inner-product function yields $GH_{\varepsilon}(IP) \in \Omega(\frac{n}{\log n})$.

3.5 Quantum garden-hose complexity

Let us consider the setting where Alice and Bob share an arbitrary entangled quantum state besides their water pipes. Depending on their respective inputs x and y, they can perform local quantum measurements on their parts of the entangled state and wire up the pipes depending on the outcomes of these measurements. We denote the resulting *quantum garden-hose complexity* with $GH^{Q}(f)$ in the deterministic case and with $GH_{\varepsilon}^{Q}(f)$ if errors are allowed.

With the same reasoning as in Proposition 3.3.12, we get that lower bounds on the entanglement-assisted one-way communication complexity carry over to the quantum garden-hose complexity (up to a logarithmic factor).

3.5.1. Proposition. *For* $\varepsilon \geq 0$*, let* $Q_{\varepsilon}^{1}(f)$ *denote the minimum cost of an entanglement-assisted one-way communication protocol which computes* f *with an error* ε*. Then,* $Q_{\varepsilon}^{1}(f) \leq GH_{\varepsilon}^{Q}(f)\log(GH_{\varepsilon}^{Q}(f))$.

For instance, the lower bound on the bounded-error one-round quantum communication complexity $Q_{\varepsilon}^{1}(IP) \in \Omega(n)$ which follows from results in [CDNT98] gives $GH_{\varepsilon}^{Q}(IP) \in \Omega(n/\log n)$. For the disjointness function, $Q_{\varepsilon}^{1}(DISJ) \in \Omega(\sqrt{n})$ from [Raz03] implies $GH_{\varepsilon}^{Q}(DISJ) \in \Omega(\sqrt{n}/\log n)$.

Here we present partial functions which give a separation between the quantum and classical garden-hose complexity in the deterministic and in the randomized setting. This shows that adding randomness or shared quantum correlations can make the garden-hose model more powerful. The proofs are direct translations of earlier separations in communication complexity by Buhrman, Cleve and Wigderson [BCW98] and Gavinsky, Kempe, Kerenidis, Raz and de Wolf [GKK^{+}07].

3.5.2. Theorem. *There exist partial Boolean functions* f *and* g *such that*

1. $GH^{Q}(f) \in O(\log n)$ *and* $GH(f) \in \Omega(\frac{n}{\log n})$,
2. $GH_{\varepsilon}^{Q}(g) \in O(\log n)$ *and* $GH_{\varepsilon}(g) \in \Omega(\frac{\sqrt{n}}{\log n})$.

3.5.1 Deterministic setting

Using techniques from [BCW98], we show a separation between the garden-hose model and the quantum garden-hose model in the deterministic setting for the function EQ', defined as:

$$EQ'(x,y) = \begin{cases} 1 & \text{if } \Delta(x,y) = 0\,, \\ 0 & \text{if } \Delta(x,y) = n/2\,, \end{cases}$$

where $\Delta(x,y)$ denotes the Hamming distance between two n-bit strings x and y. We show that the zero-error quantum garden-hose complexity of EQ' is logarithmic in the input length.

3.5.3. Theorem. *$GH^Q(EQ') \in O(\log n)$.*

Proof. Alice and Bob start with the fully entangled quantum state of $\log n$ qubits, i.e. with $\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1} |i\rangle|i\rangle$. Counting indices of the input bits from 0 to $n-1$, Alice gives a phase of -1 to state $|i\rangle$ whenever $x_i = 0$ and Bob does the same thing with his half when the bit $y_i = 0$, yielding the state

$$\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1}(-1)^{x_i+y_i}|i\rangle|i\rangle\,.$$

After both Alice and Bob perform a Hadamard transformation on their qubits, we obtain

$$\frac{1}{n\sqrt{n}}\sum_i\sum_{a,b}(-1)^{x_i+y_i}(-1)^{a\cdot i}(-1)^{b\cdot i}|a\rangle|b\rangle\,.$$

So the probability $p_{a,b}$ of obtaining outcome a, b when measuring in the computational basis is

$$p_{a,b} = \frac{1}{n^3}\left|\sum_i(-1)^{x_i+y_i+(a+b)\cdot i}\right|^2$$

If $x = y$, then $p_{a,b} = 0$ wherever $a \neq b$. If $\Delta(x,y) = n/2$, then $p_{a,b} = 0$ wherever $a = b$. It follows that $EQ'(x,y) = EQ(a,b)$ — determining the equality of the n-bit strings x and y is equivalent to computing the equality of the $\log(n)$-bit strings a and b. The garden-hose protocol for equality needs a number of pipes that is linear in the input size. After the quantum steps above, Alice and Bob can use $O(\log n)$ water pipes to compute $EQ(a,b)$. □

We can also show that the deterministic classical garden-hose complexity has an almost-linear lower bound.

3.5.4. Theorem. *$GH(EQ') \in \Omega(\frac{n}{\log n})$*

Proof. Theorem 1.7 of [BCW98] shows that the zero-error classical communication complexity of EQ' is lower bounded by $\Omega(n)$. The statement then follows from Proposition 3.3.12.

□

3.5.2 Randomized setting

The Noisy Perfect Matching problem (NPM) is a variant of the Boolean Hidden Matching introduced in [GKK$^+$07] where they prove an exponential gap between the classical one-way communication complexity and the quantum one-way communication complexity of NPM. We adapt the given quantum one-way protocol to our setting, showing that the quantum garden-hose complexity is only logarithmic. This gives a separation between the classical and quantum garden-hose complexity of a partial function in the randomized setting.

The NPM problem is described as follows:[7]

Alice's input: $x \in \{0,1\}^{2n}$.

Bob's input: a perfect matching M on $\{1,\dots,2n\}$ and a string $w \in \{0,1\}^n$. The matching M consists of n edges, $e_1 = (i_1, j_1), \dots, e_n = (i_n, j_n)$.

Promise: $\exists b \in \{0,1\}$ such that $\Delta(M \cdot x \oplus b^n, w) \leq n/3$, where $\Delta(\cdot,\cdot)$ is the Hamming distance and the k-th bit of the n-bit string $M \cdot x$ equals $x_{i_k} \oplus x_{j_k}$.

Function value: b.

Informally, the question asked is whether the parity on the edges of M, where the vertices are entries of x, is close to the parities specified by w, or not.

3.5.5. THEOREM. $GH^Q(\text{NPM}) \in O(\log n)$.

Proof. Alice and Bob use $\log(2n)$ EPR pairs as quantum state $|\psi\rangle = \frac{1}{\sqrt{2n}} \sum_{i=0}^{2n-1} |i\rangle|i\rangle$. Alice inserts her input bits $x = x_0 \dots x_{2n-1}$ as phases of the shared superposition, yielding the shared state

$$\frac{1}{\sqrt{2n}} \sum_{i=0}^{2n-1} (-1)^{x_i} |i\rangle_A |i\rangle_B \,.$$

Bob performs the following measurement: he uses projectors $P_k = |i_k\rangle\langle i_k|_B + |j_k\rangle\langle j_k|_B$ corresponding to the n edges. As they form a perfect matching, we have $\sum_{k=1}^{n} P_k = I$ and $P_k P_{k'} = \delta_{kk'} P_k$, so $\{P_k\}_k$ is a valid orthogonal measurement. Let us denote Bob's measurement outcome by ℓ. Setting $i := i_\ell$ and $j := j_\ell$, the post-measurement state is

$$(-1)^{x_i} |i\rangle_A |i\rangle_B + (-1)^{x_j} |j\rangle_A |j\rangle_B \,.$$

Alice then performs a Hadamard transform $H^{\otimes 2n} \otimes I$ on her part of the state, resulting in

$$\sum_{a=0}^{2n-1} |a\rangle_A \Big[(-1)^{x_i + a \cdot i} |i\rangle_B + (-1)^{x_j + a \cdot j} |j\rangle_B \Big] \,.$$

---

[7] For this example, we deviate from the earlier convention of giving two n-bit strings as input to the players.

Alice measures her register in the computational basis and obtains outcome a. Bob performs a Hadamard gate on basis states $|i\rangle_B$ and $|j\rangle_B$, that is, $H_{i,j} = \frac{1}{2}(|i\rangle\langle i|_B + |i\rangle\langle i|_B + |j\rangle\langle j|_B - |j\rangle\langle j|_B)$, resulting in the state

$$|a\rangle_A \biggl(\frac{1}{2}\Bigl[(-1)^{x_i+a\cdot i} + (-1)^{x_j+a\cdot j}\Bigr]|i\rangle_B \\ + \frac{1}{2}\Bigl[(-1)^{x_i+a\cdot i} - (-1)^{x_j+a\cdot j}\Bigr]|j\rangle_B \biggr).$$

and measures in the computational basis. He gets outcome i if and only if $x_i \oplus a\cdot i = x_j \oplus a \cdot j$ which is equivalent to $x_i \oplus x_j = a \cdot (i \oplus j)$. In case $x_i \oplus x_j \neq a \cdot (i \oplus j)$, Bob gets outcome j.

After the measurements, Alice and Bob perform the garden-hose protocol for the inner-product function described in Section 3.3.4 with a and $i \oplus j$ as their respective inputs. The protocol can be easily adapted so that at the end of it, the water will be in one particular pipe (known to Bob) on Bob's side if $a \cdot (i \oplus j) = 0$, let us call this pipe 0-pipe. The water will be in another "1-pipe" (known to Bob) if $a \cdot (i \oplus j) = 1$. Furthermore, Bob knows from his second measurement outcome if they are computing $x_i \oplus x_j$ or $x_i \oplus x_j \oplus 1$. In the first case, Bob looks at the ℓ-th bit of w and leaves the 0-pipe open if $w_\ell = 1$ and routes the 1-pipe to Alice, and if $w_\ell = 0$ he keeps the 1-pipe open and sends back the 0-pipe. This strategy computes the function value $w_\ell \oplus x_i \oplus x_j$, with ℓ uniformly random in $\{1, \ldots, n\}$. The promise guarantees that it gives the correct value b with probability at least $\frac{2}{3}$. The second case (when Bob knows that $a \cdot (i \oplus j) \neq x_i \oplus x_j$) is handled by the "inverse" strategy. □

3.5.6. Theorem. $GH_\varepsilon(\text{NPM}) \in \Omega(\frac{\sqrt{n}}{\log n})$.

Proof. Combining the lower bound on the classical one-way communication complexity from [GKK+07] of $\Omega(\sqrt{n})$ with Proposition 3.4.5 gives the statement. □

3.6 Lower bounds on quantum resources to perfectly attack $\mathsf{PV}_{\text{qubit}}$

In Section 3.6.3, we show that for a function that is injective for Alice or injective for Bob, according to Definition 3.3.5, the dimension of the quantum state the adversaries need to handle (including possible quantum communication between them) in order to attack protocol $\mathsf{PV}_{\text{qubit}}$ perfectly has to be of order at least linear in the classical input size n. In other words, they require at least a logarithmic number of qubits in order to successfully attack $\mathsf{PV}_{\text{qubit}}$.

3.6.1. Theorem. *Let f be injective for Bob. Assume that Alice and Bob perform a perfect attack on protocol $\mathsf{PV}_{\text{qubit}}$. Then, the dimension d of the overall state (including the quantum communication) is in $\Omega(n)$.*

In the last subsection, we show that there exist functions for which perfect attacks on $\mathsf{PV}_{\text{qubit}}$ require the adversaries to handle a polynomial amount of qubits.

3.6.2. THEOREM. *For any starting state $|\psi\rangle$ of dimension d, there exists a Boolean function f on inputs $x, y \in \{0,1\}^n$ such that any perfect attack on* $\mathsf{PV}_{\text{qubit}}$ *requires d to be exponential in n.*

These results can be seen as first steps towards establishing the desired relation between classical difficulty of honest actions and quantum difficulty of the actions of dishonest players. We leave as future work the generalization of these lower bounds to the more realistic case of imperfect attacks and also to more relevant quantities like some entanglement measure between the players (instead of the dimension of their shared state).

We show that for a function that is injective for Alice or injective for Bob (according to Definition 3.3.5), the dimension of the state the adversaries need to handle (including possible quantum communication between them) in order to attack protocol $\mathsf{PV}_{\text{qubit}}$ perfectly has to be of order at least linear in the classical input size n. We start by showing two lemmas. The actual bound is shown in Section 3.6.3.

In Section 3.6.4 we show that there exist functions for which perfect attacks on $\mathsf{PV}_{\text{qubit}}$ requires the adversaries to handle a polynomial amount of qubits. Do note that not allowing the adversaries any error is not realistic for cryptographic purposes: the proofs in this section are only a stepping stone for showing a trade-off between the amount of classical information and the size of the quantum state needed to execute a realistic attack on the protocol.

3.6.1 Localized qubits

Assume we have two bipartite states $|\psi^0\rangle$ and $|\psi^1\rangle$ with the property that $|\psi^0\rangle$ allows Alice to locally extract a qubit and $|\psi^1\rangle$ allows Bob to locally extract the same qubit. Intuitively, these two states have to be different.

More formally, we assume that both states consist of five registers $R, A, \tilde{A}, B, \tilde{B}$ where R, A, B are one-qubit registers and $\tilde{A}$ and $\tilde{B}$ are arbitrary. We assume that there exist local unitary transformations $U_{A\tilde{A}}$ acting on registers $A\tilde{A}$ and $V_{B\tilde{B}}$ acting on $B\tilde{B}$ such that[8]

$$U_{A\tilde{A}}\left|\psi^0\right\rangle_{RA\tilde{A}B\tilde{B}} = |\beta\rangle_{RA} \otimes |P\rangle_{\tilde{A}B\tilde{B}} \tag{3.1}$$

$$V_{B\tilde{B}}\left|\psi^1\right\rangle_{RA\tilde{A}B\tilde{B}} = |\beta\rangle_{RB} \otimes |Q\rangle_{A\tilde{A}\tilde{B}}\,, \tag{3.2}$$

where $|\beta\rangle_{RA} := (|00\rangle_{RA} + |11\rangle)_{RA})/\sqrt{2}$ denotes an EPR pair on registers RA and $|P\rangle_{\tilde{A}B\tilde{B}}$ and $|Q\rangle_{A\tilde{A}\tilde{B}}$ are arbitrary pure states.

---

[8] We always assume that these transformations act as the identities on the registers we do not specify explicitly.

3.6.3. Lemma. *Let* $|\psi^0\rangle, |\psi^1\rangle$ *be states that fulfill* (3.1) *and* (3.2). *Then,*

$$\big|\langle\psi^0|\psi^1\rangle\big| \leq 1/2\,.$$

Proof. Multiplying both sides of (3.1) with $U^\dagger_{A\tilde{A}}$ and multiplying (3.2) with $V^\dagger_{B\tilde{B}}$, we can write

$$\begin{aligned}\big|\langle\psi^0|\psi^1\rangle\big| &= \big|\langle\beta|_{RA}\langle P|_{\tilde{A}B\tilde{B}}\,U_{A\tilde{A}}\,V^\dagger_{B\tilde{B}}\,|\beta\rangle_{RB}|Q\rangle_{A\tilde{A}\tilde{B}}\big| \\ &= \big|\langle\beta|_{RA}\langle P'|_{\tilde{A}B\tilde{B}}|\beta\rangle_{RB}|Q'\rangle_{A\tilde{A}\tilde{B}}\big| \\ &= \big|\langle P'|_{\tilde{A}B\tilde{B}}\langle\beta|_{RA}|\beta\rangle_{RB}|Q'\rangle_{A\tilde{A}\tilde{B}}\big|\,,\end{aligned}$$

where we used that $U_{A\tilde{A}}$ and $V_{B\tilde{B}}$ commute and defined $|P'\rangle_{\tilde{A}B\tilde{B}} := V_{B\tilde{B}}|P\rangle_{\tilde{A}B\tilde{B}}$ and $|Q'\rangle_{A\tilde{A}\tilde{B}} := U_{A\tilde{A}}|Q\rangle_{A\tilde{A}\tilde{B}}$. The last equality is just rearranging terms that act on different registers.

Note that writing out the partial inner product between $|\beta\rangle_{RA}$ and $|\beta\rangle_{RB}$ gives

$$\langle\beta|_{RA}|\beta\rangle_{RB} = \frac{1}{2}\Big(\langle 0|_A|0\rangle_B + \langle 1|_A|1\rangle_B\Big)\,,$$

where the operator in the parenthesis "transfers" a qubit from register A to register B. Hence,

$$\begin{aligned}\big|\langle\psi^0|\psi^1\rangle\big| &= \big|\langle P'|_{\tilde{A}B\tilde{B}}\frac{1}{2}\Big(\langle 0|_A|0\rangle_B + \langle 1|_A|1\rangle_B\Big)|Q'\rangle_{A\tilde{A}\tilde{B}}\big| \\ &= \frac{1}{2}\cdot\big|\langle P'|_{\tilde{A}B\tilde{B}}|Q'\rangle_{B\tilde{A}\tilde{B}}\big| \\ &\leq \frac{1}{2}\,,\end{aligned}$$

where the last step follows from the fact that the inner product between any two unit vectors on the same registers can be at most 1. □

3.6.2 Squeezing many vectors in a small space

For the sake of completeness, we reproduce here an argument similar to [NC00, Section 4.5.4] about covering the state space of dimension d with patches of radius ε.

3.6.4. Lemma. *Let* $\mathcal{B}$ *be a set of* 2^n *distinct unit vectors in a complex Hilbert space of dimension* d*, with pairwise absolute inner product at most* $1/2$. *Then, the dimension* d *has to be in* $\Omega(n)$.

Proof. For any two vectors $|v\rangle, |w\rangle$, we can rotate the space such that $|v\rangle = |0\rangle$ and $|w\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle$ for two orthogonal vectors $|0\rangle$ and $|1\rangle$. The *Euclidean distance* between $|v\rangle$ and $|w\rangle$ can be expressed as

$$\begin{aligned}\Big| |v\rangle - |w\rangle \Big| &= |(1-\cos\theta)|0\rangle - \sin\theta|1\rangle| \\ &= \sqrt{(1-\cos\theta)^2 + \sin^2\theta} \\ &= \sqrt{1 - 2\cos\theta + \cos^2\theta + \sin^2\theta} \\ &= \sqrt{2}\sqrt{1-\cos\theta}\,.\end{aligned}$$

If $|v\rangle$ and $|w\rangle$ have absolute inner product at most $1/2$, we have that $|\cos\theta| \leq 1/2$ and hence $\Big| |v\rangle - |w\rangle \Big| \geq 1$. Therefore, the vectors in $\mathcal{B}$ have pairwise Euclidean distance at least 1. The set of unit vectors $|w\rangle$ with Euclidean distance at most δ from $|v\rangle$ is called *patch of radius δ around $|v\rangle$*. It follows that patches of radius $1/2$ around every vector in the set $\mathcal{B}$ do not overlap.

The space of all d-dimensional state vectors can be regarded as the real unit $(2d-1)$-sphere, because the vector has d complex amplitudes and hence $2d$ real degrees of freedom with the restriction that the sum of the squared amplitudes is equal to 1. Notice that the Euclidean distance between complex vectors $|v\rangle$, $|w\rangle$ remains unchanged if we regard these vectors as points of the real unit $(2d-1)$-sphere.

The surface area of a patch of radius $1/2$ near any vector is lower bounded by the volume of a $(2d-2)$-sphere of radius ε where ε is a constant slightly less than $1/2$.[9]. We use the formula $S_k(r) = 2\pi^{(k+1)/2} r^k / \,\Gamma((k+1)/2)$ for the surface area of a k-sphere of radius r, and $V_k(r) = 2\pi^{(k+1)/2} r^{k+1} / [(k+1)\,\Gamma((k+1)/2)]$ for the volume of a k-sphere of radius r. The total surface area of all patches, which is at least $2^n \cdot V_{2d-2}(\varepsilon)$, is not more than the total surface of the whole sphere $S_{2d-1}(1)$. Inserting the formulas, we get

$$2^n \cdot 2\pi^{d-\frac{1}{2}} \frac{\varepsilon^{2d-1}}{(2d-1)\,\Gamma(d-\frac{1}{2})} \leq 2\pi^d \frac{1}{\Gamma(d)}$$

Using the fact that $\frac{\Gamma(d-\frac{1}{2})}{\Gamma(d)} \leq \frac{1}{d}$, we conclude that

$$2^n \leq \sqrt{\pi}(2 - \frac{1}{d})\varepsilon^{-(2d-1)} \leq 2\sqrt{\pi}\varepsilon^{-(2d-1)}\,.$$

As $\varepsilon < 1/2$, we obtain that d has to be in $\Omega(n)$. □

3.6.3 The lower bound

We consider perfect attacks on protocol $\mathsf{PV}_{\text{qubit}}$ from Figure 3.1. We allow the players one round of simultaneous quantum communication which we model as follows. Let $|\psi\rangle_{RA\tilde{A}A_C B\tilde{B}B_C}$ be the pure state after Alice received the EPR half from the verifier. The one-qubit register R holds the verifier's half of the EPR pair, the one-qubit register A contains Alice's other half of the EPR pair, the register $\tilde{A}$ is Alice's part of the pre-shared entangled state and the register A_C holds the qubits that will be communicated to Bob. The registers $B\tilde{B}B_C$ belong to Bob where B holds one qubit and $\tilde{B}$ is Bob's part of the entangled state and the B_C register will be sent to Alice. We denote by q_A the total number of qubits in registers $\tilde{A}$ and A_C and by q_B the total number of qubits in $\tilde{B}$ and B_C. The overall state is thus a unit vector in a complex Hilbert space of dimension $d := 2^{2+q_A+1+q_B}$.

---

[9] The patch is a “bent” version of this volume.

In the first step of their attack, Alice performs a unitary transform U^x depending on her classical input x on her registers $A\tilde{A}A_C$. Similarly, Bob performs a unitary transform V^y depending on y on registers $B\tilde{B}B_C$. After the application of these transforms, the communication registers A_C and B_C and the classical inputs x and y are exchanged. A final unitary transform (performed either by Alice or Bob) depending on both x, y "unveils" the qubit either in Alice's register A or in Bob's register B.

3.6.5. Theorem. *Let f be injective for Bob. Assume that Alice and Bob perform a perfect attack on protocol $\mathsf{PV}_{\text{qubit}}$. Then, the dimension d of the overall state (including the quantum communication) is in $\Omega(n)$.*

Proof. We assume that the player's actions are unitary transforms as described before the theorem.

We investigate the set $\mathcal{B}$ of overall states after Bob performed his operation, but *before* Alice acts on the state. These states depend on Bob's input $y \in \{0,1\}^n$,

$$\mathcal{B} := \left\{ V^y_{B\tilde{B}B_C} |\psi\rangle_{RA\tilde{A}A_C B\tilde{B}B_C} : y \in \{0,1\}^n \right\}.$$

We claim that for any two different n-bit strings $y \neq y'$, the corresponding two vectors $V^y|\psi\rangle$ and $V^{y'}|\psi\rangle$ in $\mathcal{B}$ have an absolute inner product of at most $1/2$.

Due to the injectivity of f, there exists an input x for Alice such that $f(x,y) \neq f(x,y')$. Applying Alice's unitary transform U^x to both vectors does not change their inner product, i.e.

$$|\langle\psi|(V^y)^\dagger V^{y'}|\psi\rangle| = |\langle\psi|(V^y)^\dagger (U^x)^\dagger U^x V^{y'}|\psi\rangle| .$$

As $f(x,y) \neq f(x,y')$, the qubit has to end up on different sides. Formally, there exist unitary transforms $K_{A\tilde{A}B_C}$ and $L_{B\tilde{B}A_C}$ that "unveil" the qubit in register A or B respectively. Hence, we can apply Lemma 3.6.3 to prove the claim that the two vectors $V^y|\psi\rangle$ and $V^{y'}|\psi\rangle$ have an absolute inner product of at most $1/2$. In particular, all of the vectors in $\mathcal{B}$ are distinct. Applying Lemma 3.6.4 yields the theorem. □

3.6.4 Functions for which perfect attacks need a large space

Using similar arguments as above, we can show the existence of functions for which perfect attacks require polynomially many qubits.

3.6.6. Theorem. *For any starting state $|\psi\rangle$ of dimension d, there exists a Boolean function on inputs $x, y \in \{0,1\}^n$ such that any perfect attack on $\mathsf{PV}_{\text{qubit}}$ requires d to be exponential in n.*

We believe that the statement with the reversed order of quantifiers is true as well (but our current proof does not suffice for this purpose), so that we can guarantee the existence of one particular function (independent of the starting state) for which perfect attacks require large states.

Proof sketch. We consider covering the sphere with K patches of vectors whose pairwise absolute inner product is larger than $\frac{\sqrt{3}}{2}$ (which corresponds to an Euclidean distance of $\varepsilon = \sqrt{2}\sqrt{1+\sqrt{3}/2} \approx 0.52$). This partitioning also induces a partitioning on all possible unitary operations of Alice and Bob. We say that two actions A and A' are in the same patch if they take the starting state $|\psi\rangle$ to the same patch. In other words, if two actions are in the same patch then

$$\left|\langle\psi|A'^{\dagger}A|\psi\rangle\right| \geq \frac{\sqrt{3}}{2}.$$

Claim. Given two actions of Alice A, A' coming from the same patch i, and two actions of Bob B, B' coming from the same patch j, the inner product between $BA|\psi\rangle$ and $B'A'|\psi\rangle$ has magnitude at least $\frac{1}{2}$.

Proof of the claim. Since Alice and Bob act on different parts of the state, their actions commute. Write $|\psi_A\rangle := A'^{\dagger}A|\psi\rangle$ and $|\psi_B\rangle := B^{\dagger}B'|\psi\rangle$. Then the inner product can be written as

$$\langle\psi|A'^{\dagger}B'^{\dagger}BA|\psi\rangle = \langle\psi|B'^{\dagger}BA'^{\dagger}A|\psi\rangle = \langle\psi_B|\psi_A\rangle$$

Note that

$$\left|\langle\psi|\psi_A\rangle\right| = \left|\langle\psi|A'^{\dagger}A|\psi\rangle\right| \geq \frac{\sqrt{3}}{2},$$

so the angle θ between $|\psi_A\rangle$ and $|\psi\rangle$ is at most $\arccos\frac{\sqrt{3}}{2} = \frac{\pi}{6}$. The same holds for the angle between $|\psi_B\rangle$ and $|\psi\rangle$. We can upper bound the total angle between $|\psi_A\rangle$ and $|\psi_B\rangle$ by the sum of these angles, giving a total angle of at most $\frac{\pi}{3}$. This corresponds to a lower bound on the inner product of $\cos\frac{\pi}{3} = \frac{1}{2}$. □

So there exists no pair of combined actions AB and $A'B'$, with A and A' in patch i and B and B' in patch j, such that the qubit ends up on Alice's side for

AB and on Bob's side for $A'B'$. Therefore, the combination of i and j completely determines the destination of the qubit and hence the output of the function. If K denotes the number of patches, then there are K^{2^n} possible strategies for Alice and K^{2^n} possible strategies for Bob. Hence, the number of combined strategies (possibly resulting in different functions) is at most $K^{2 \cdot 2^n}$.

It is shown in [NC00, Section 4.5.4] that we need at least $K = \Omega(\frac{1}{\varepsilon^{d-1}})$ patches. Using the same counting argument as in Proposition 3.3.9, we have that

$$2^{2^{2n}} \geq \Omega\left(\frac{1}{\varepsilon^{(d-1)2 \cdot 2^n}}\right),$$

from which follows that for some function, d has to be exponential in n. □

3.7 Conclusion and open questions

The garden-hose model is a new model of communication complexity. We connected functions with polynomial garden-hose complexity to a newly defined class of log-space computations with local pre-processing. Alternatively, the class $\mathrm{L}_{(2)}$ can also be viewed as the set of functions which can be decided in the simultaneous-message-passing (SMP) model where the referee is restricted to log-space computations. Many open questions remain. Can we find better upper and lower bounds for the garden-hose complexity of the studied functions? The constructions given in [Spe11] still leave a polynomial gap between lower and upper bounds for many functions. It would also be interesting to find an explicit function for which the garden-hose complexity is provably super-linear or even exponential, the counting argument in Proposition 3.3.9 only showed the existence of such functions. It is possible to extend the basic garden-hose model in various ways and consider settings with more than two players, non-Boolean functions or multiple water sources. Furthermore, it is interesting to relate our findings to recent results about space-bounded communication complexity [BCP+13].

Garden-hose complexity is a tool for the analysis of a specific scheme for position-based quantum cryptography. This scheme requires the honest prover to work with only a single qubit, while the dishonest provers potentially have to manipulate a large quantum state, making it an appealing scheme to further examine. The garden-hose model captures the power of attacks that only use teleportation, giving upper bounds for the general scheme, and lower bounds when restricted to these attacks.

An interesting additional restriction on the garden-hose model would involve limiting the computational power of Alice and Bob. For example to polynomial time, or to the output of quantum circuits of polynomial size. Bounding not only the amount of entanglement, but also the amount of computation with a realistic limit might yield stronger security guarantees for the cryptographic schemes.

Chapter 4

Instantaneous non-local computation of low T-depth quantum circuits

Instantaneous non-local quantum computation requires multiple parties to jointly perform a quantum operation, using pre-shared entanglement and a single round of simultaneous communication. We study this task for its close connection to position-based quantum cryptography, but it also has natural applications in the context of foundations of quantum physics and in distributed computing. The best known general construction for instantaneous non-local quantum computation requires a pre-shared state which is exponentially large in the number of qubits involved in the operation, while efficient constructions are known for very specific cases only.

We partially close this gap by presenting new schemes for efficient instantaneous non-local computation of several classes of quantum circuits, using the Clifford+T gate set. Our main result is a protocol which uses entanglement exponential in the T-depth of a quantum circuit, able to perform non-local computation of quantum circuits with a (poly-)logarithmic number of layers of T gates with quasi-polynomial entanglement. Our proofs combine ideas from blind and delegated quantum computation with the garden-hose model, a combinatorial model of communication complexity which was recently introduced as a tool for studying certain schemes for quantum position verification. As application of our results, we also present an efficient attack on a recently-proposed scheme for position verification by Chakraborty and Leverrier.

The results in this chapter are also available as the following paper:

- [Spe16] Florian Speelman. Instantaneous non-local computation of low T-depth quantum circuits. In *11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, pages 9:1–9:24, 2016.

4.1 Introduction

We continue our study of position-based quantum cryptography by focusing on the task of *instantaneous non-local quantum computation*, and present new protocols to efficiently perform this task for specific classes of quantum circuits. Our main motivation comes from position-based quantum cryptography, where previous attacks on schemes for position-based quantum cryptography have taken either of two forms:

First results on quantum position-based cryptography involved attacks on specific proposals for schemes, such as the attacks by Lau and Lo [LL11], those by Kent, Munro and Spiller [KMS11], and the attack on Beigi and König's scheme using mutually-unbiased-bases [Spe11]. The garden-hose model that we studied in Chapter 3 can also be seen as an efficient attack on a concrete class of single-qubit schemes [BFSS13]. Described as 'fast protocols for bipartite unitary operators', Yu, Griffiths and Cohen [YGC12, Yu11] give protocols that, although not directly inspired by position-based quantum cryptography, can be translated to our setting.

On the other hand, as we mentioned earlier in Section 1.1, Buhrman et al. [BCF$^+$11] constructed a general attack which treats the quantum functionality of the protocol to be attacked as a black box. For a protocol which uses a message of n qubits, the entanglement consumption of this attack is around $2^{\log(\frac{1}{\varepsilon})2^{4n}}$ EPR pairs, doubly exponential in n. Here ε represents the probability that the attack does not succeed. The construction of Buhrman et al. was based on a protocol for 'instantaneous non-local measurement' by Vaidman [Vai03, CCJP10]. Beigi and König [BK11] later constructed a more efficient general attack, using port-based teleportation – a new teleportation method introduced by Ishizaka and Hiroshima [IH08, IH09]. The improved attack uses $O(n\frac{2^{8n}}{\varepsilon^2})$ EPR pairs, still an exponential dependence on n.

These protocols were able to solve the following task, of which we here give an intuitive description. We present a more precise general definition of this task in Section 4.3. Given a constant $\varepsilon \geq 0$ and an n-qubit quantum operation[1] U, where n is a natural number. Two players, Alice and Bob, receive an arbitrary input state ρ_{AB} of n qubits, with the players receiving $n/2$ qubits each. After a single round of simultaneous quantum[2] communication, the players must output a state ε-close to $U\rho_{AB}U^\dagger$. Alice outputs the first $n/2$ qubits of the state and Bob outputs the other $n/2$ qubits. We define $\mathrm{INQC}_\varepsilon(U)$ as the smallest number of

---

[1] Our constructions only consider unitaries given by quantum circuits, but the task naturally extends to more general quantum operations. The motivation for Vaidman's original scheme [Vai03], which formed the basis of Buhrman et al.'s construction, was to instantaneously perform a non-local measurement. Our constructions can also be applied to that case, by writing the measurement as a unitary operation followed by a measurement in the computational basis.

[2] Since restriction to classical communication is not necessarily dictated by the application in position-based quantum cryptography, we allow quantum communication. All presented protocols work equally well when all messages are classical instead.

EPR pairs that the players have to share at the start of a protocol which performs this task. INQC(U) is used as a shorthand for $\mathrm{INQC}_0(U)$, a protocol which works with no error.

Any protocol for quantum position verification has to be easy to implement for honest parties, to be usable in practice. The hardness of a quantum computation is commonly analyzed in terms of properties of the associated *quantum circuit*, such as the size or depth.

In this chapter, we will partially bridge the gap between efficient specific constructions for instantaneous non-local computation and expensive general ones, by constructing a protocol for non-local computation of a unitary transformation U such that the entanglement use of the protocol depends on the quantum circuit which describes U.

In particular, we will write quantum circuits over the Clifford+T gate set, and create a protocol using entanglement exponential in the *T-count*. We will also create a protocol that uses an amount of entanglement which is exponential in the *T-depth* of the circuit, where the number of qubits n is part of the base of the dependence. Even though this is a quickly-growing dependence, for circuits of constant T-depth this amounts to a polynomial dependence on n, unlike any earlier construction. For circuits of polylogarithmic T-depth we obtain an amount of entanglement which is quasi-polynomial in n, i.e. a dependence of the form $2^{(\log n)^c}$ for some constant c. Note that the depth and size of the quantum circuit can be much higher than its T-depth—we allow an arbitrary number of gates from the Clifford group besides the limited number of T gates. Our results therefore imply new efficient attacks on any scheme for position-verification where the action of the honest party can be written as a low T-depth quantum circuit.

Linking blind quantum computation and instantaneous non-local quantum computation was first considered by Broadbent[3] [Bro15b], who considered a setting where the parties have access to non-local boxes – correlations even stronger than those allowed by quantum mechanics. The techniques we use are also based on delegated and blind quantum computation [Chi05, AS06, DNS10, FBS+14, Bro15a] and results on computation via teleportation [GC99], but we combine them with new ideas from the *garden-hose model* [BFSS13, KP14]. We introduced the garden-hose model in the previous chapter as a combinatorial model for communication complexity, inspired by attacks on a very specific class of schemes for position verification.

In this chapter we prove two main theorems, each improving on the entanglement consumption of the best known previous constructions for non-local instantaneous quantum computation for specific circuits. From now on, whenever we write ‘quantum circuit’, we will always mean a quantum circuit that only uses gates that generate the Clifford group, together with T gates. Additionally, we use our proof method to construct a new attack on a scheme for position verification which was recently proposed by Chakraborty and Leverrier [CL15].

---

[3] These results were first available as privately-circulated notes in December 2011, and were made available online in December 2015.

Theorem 4.4.1 Any n-qubit Clifford+T quantum circuit C which has at most k T gates has a protocol for instantaneous non-local computation using $O(n2^k)$ EPR pairs.

Theorem 4.6.1 Any n-qubit quantum circuit C using the Clifford+T gate set which has T-depth d, has a protocol for instantaneous non-local computation using $O(\,(68n)^d\,)$ EPR pairs.

We can apply the construction of Theorem 4.4.1 to the Clifford hierarchy, also called the Gottesman–Chuang hierarchy [GC99], to obtain the following result. Since the dependence on n is exponential, the (error-less) Proposition 4.4.2 will only be a qualitative improvement over Beigi and König's port-based teleportation construction when both n and the level k are small.

Proposition 4.4.2 If U is an n-qubit operation in the k-th level of the Clifford hierarchy, where Alice receives $n/2$ qubits and Bob receives $n/2$ qubits, then $\mathrm{INQC}(U) \leq O(n4^{nk})$.

The main technical tool we use in the proof of our depth-dependent construction is the following lemma, which is able to remove a conditionally-applied gate from the Clifford group without any communication – at an entanglement cost which scales with the garden-hose complexity of the function which describes the condition.

Lemma 4.5.1 Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function known to all parties, and let $GH(f)$ be the garden-hose complexity of the function f. Assume Alice has a single qubit with state $\mathrm{P}^{f(x,y)}|\psi\rangle$, for binary strings $x, y \in \{0,1\}^n$, where Alice knows the string x and Bob knows y. The following two statements hold:

1. There exists an instantaneous protocol without any communication which uses $2GH(f)$ pre-shared EPR pairs after which a chosen qubit of Alice is in the state $\mathrm{X}^{g(\hat{x},\hat{y})}\mathrm{Y}^{h(\hat{x},\hat{y})}|\psi\rangle$. Here $\hat{x}$ depends only on x and the $2GH(f)$ bits that describe the measurement outcomes of Alice, and $\hat{y}$ depends on y and the measurement outcomes of Bob.

2. The garden-hose complexities of the functions g and h are at most linear in the garden-hose complexity of the function f. More precisely, $GH(g) \leq 4GH(f) + 1$ and $GH(h) \leq 11GH(f) + 2$.

Chakraborty and Leverrier [CL15] recently proposed a protocol for quantum position verification on the interleaved multiplication of unitaries, the Interleaved Product protocol. They show that all known attacks, applied to this protocol, require entanglement exponential in the number of terms t in the product. As application of Lemma 4.5.1, we present an attack on their proposed protocol which has entanglement cost polynomial in t and the number of qubits n. The new attack requires an amount of entanglement which scales as $(\frac{t}{\varepsilon})^{O(1)}$ per qubit, and for each qubit succeeds with probability at least $1-\varepsilon$.

4.2 Preliminaries

4.2.1 The Pauli matrices and the Clifford group

Recall from Chapter 2 that the single-qubit *Pauli matrices* are $\mathrm{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\mathrm{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, $\mathrm{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, and the identity $\mathrm{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. A *Pauli operator* on an n-qubit state is the tensor product of n one-qubit Pauli matrices, the group of n qubit Pauli operators[4] is $\mathcal{P} = \{\sigma_1 \otimes \cdots \otimes \sigma_n \mid \forall j : \sigma_j \in \{I, X, Y, Z\}\} \times \{\pm 1, \pm i\}$. These are some of the simplest quantum operations and appear, for example, as corrections for standard quantum teleportation.

The *Clifford group* can be defined as those operations that take elements of the Pauli group to other elements of the Pauli group under conjugation – the *normalizer* of the Pauli group. We consider the Clifford group on n qubits, for some natural number n.

$$\mathcal{C} = \{U \in \mathcal{U}(2^n) \mid \forall \sigma : \sigma \in \mathcal{P} \implies U \sigma U^\dagger \in \mathcal{P}\} \tag{4.1}$$

Notable elements of the Clifford group are the single-qubit gates given by the Hadamard matrix

$$\mathrm{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

and the phase gate

$$\mathrm{P} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$

and the two-qubit CNOT gate given by

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

---

[4] The given definition includes a global phase, which is not important when viewing the elements as quantum gates.

Applied to arbitrary qubits, the set $\{\mathrm{H}, \mathrm{P}, \mathrm{CNOT}\}$ generates the Clifford group (up to a global phase) [Got98b]. For all these gates, we will use subscripts to indicate the qubits or wires to which they are applied; e.g. H_j is a Hadamard gate applied to the j-th wire, and $\mathrm{CNOT}_{j,k}$ is a CNOT that has wire j as control and k as target.

Even though there exist interesting quantum circuits that use only gates from the Clifford group, it is not a universal set of gates. Indeed, the Gottesman–Knill Theorem states that such a circuit can be efficiently simulated by a classical computer, something which is not known to be true for general quantum circuits [Got98a, AG04]. By extending $\mathcal{C}$ with *any* gate, we do obtain a gate-set which is universal for quantum computation [NRS01].

The gate we will use to extend the Clifford gates to a universal set is the T gate, sometimes called $\pi/8$-gate or R, defined by $\mathrm{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$. We will write all circuits using gates from the set $\{\mathrm{X}, \mathrm{Z}, \mathrm{H}, \mathrm{P}, \mathrm{CNOT}, \mathrm{T}\}$[5].

In our protocols for instantaneous non-local computation, we will alternate teleportation steps with gate operations, and therefore the interaction between the Pauli matrices and the other gates are especially important. We will make much use of the following identities, which can be easily checked by hand[6].

$$\begin{array}{ll}
\mathrm{XZ} = \mathrm{ZX} & \\
\mathrm{PZ} = \mathrm{ZP} & \mathrm{CNOT}_{1,2}(\mathrm{X} \otimes \mathrm{I}) = (\mathrm{X} \otimes \mathrm{X})\mathrm{CNOT}_{1,2} \\
\mathrm{PX} = \mathrm{XZP} & \mathrm{CNOT}_{1,2}(\mathrm{I} \otimes \mathrm{X}) = (\mathrm{I} \otimes \mathrm{X})\mathrm{CNOT}_{1,2} \\
\mathrm{HX} = \mathrm{ZH} & \mathrm{CNOT}_{1,2}(\mathrm{Z} \otimes \mathrm{I}) = (\mathrm{Z} \otimes \mathrm{I})\mathrm{CNOT}_{1,2} \\
\mathrm{HZ} = \mathrm{XH} & \mathrm{CNOT}_{1,2}(\mathrm{I} \otimes \mathrm{Z}) = (\mathrm{Z} \otimes \mathrm{Z})\mathrm{CNOT}_{1,2} \\
\mathrm{TX} = \mathrm{PXT} &
\end{array} \tag{4.2}$$

4.2.2 Key transformations from Clifford circuits

For a vector $v \in \{0,1\}^n$ and for any single-qubit operation U, we write $U^v = \bigotimes_{j=1}^n U^{v_j}$, i.e., U^v is the application of U on all qubits $j \in [n]$ for which $v_j = 1$. When Alice teleports a state $|\psi\rangle$ of n qubits to Bob, before Bob knows the teleportation correction, the state at Bob's side can be written as $\mathrm{X}^{a_x}\mathrm{Z}^{a_z}|\psi\rangle$, when we let a_x and a_z be the vectors representing the outcomes of the Bell measurements of Alice. In analogy with the the literature on assisted and blind quantum computation, we will call the teleportation measurement outcomes a_x and a_z the *key* needed to decode $|\psi\rangle$.

The specific entries of these keys will often depend on several different measurement outcomes, given by earlier steps in the protocol, and we will therefore occasionally describe them as *polynomials* over $\mathbb{F}_2$. Viewing the keys as polynomials is especially helpful in the description of the more-complicated protocol of Section 4.6.

---

[5] Technically X, P, and Z are redundant here, since they can be formed by the others as $\mathrm{P} = \mathrm{T}^2$, $\mathrm{Z} = \mathrm{P}^2$ and $\mathrm{X} = \mathrm{ZHZ}$, but we include them because of their ubiquity.

[6] Here equality is up to a global phase – which we will ignore from now on for simplicity.

For any gate from the Clifford group $U \in \mathcal{C}$, if we apply U on the encoded state, we can describe the resulting state as $U|\psi\rangle$ with a new key. That is, $UX^{a_x}Z^{a_z}|\psi\rangle = X^{\hat{a}_x}Z^{\hat{a}_z}U|\psi\rangle$ for some new 0/1 keys $\hat{a}_x, \hat{a}_z$. The transformations of the keys will have a particularly simple form. (See for example [BCL$^+$06] for a characterization of these transformations and a different application of Clifford circuit computation.)

For example, we can write the identities of Equation 4.2 in terms of key transformations. The transformations that occur when a bigger Pauli operator is applied, can then be easily found by writing the Pauli operator in terms of its generators $\{\mathrm{H}, \mathrm{P}, \mathrm{CNOT}\}$, and applying these rules one-by-one. We will write $(x_1, x_2 \mid z_1, z_2)$ as a shorthand for, respectively, the X key on the first and second qubit, and the Z key on the first and second qubit – this is a convenient notation for the pair of vectors a_x and a_z that represent these keys. All addition of these keys will be over $\mathbb{F}_2$, i.e., the + represents the binary exclusive or.

$$\begin{aligned} \mathrm{P}(x \mid z) &\to (x \mid x+z)\mathrm{P} \\ \mathrm{H}(x \mid z) &\to (z \mid x)\mathrm{H} \\ \mathrm{CNOT}_{1,2}(x_1, x_2 \mid z_1, z_2) &\to (x_1, x_1 + x_2 \mid z_1 + z_2, z_2)\mathrm{CNOT}_{1,2} \end{aligned}$$

4.2.3 Clifford+T quantum circuits, T-count and T-depth

All circuits in this chapter will be written using gates from the Clifford group, and the T gate. In several different areas of quantum information, gates from the Clifford group are ‘well-behaved’ or ‘easy’, while the other non-Clifford gates are hard – an observation which was also made, with several examples, in the recent [BJ15].

The *T-count* of a quantum circuit is defined as the number of T gates in the entire quantum circuit. The *T-depth* is the number of layers of T gates, when viewing the circuit as alternating between Clifford gates and a layer of simultaneous T gates. See for example Figure 4.6.

Given a quantum operation, it is not always obvious what is the best circuit in terms of T-count or T-depth. Recent work gave algorithms for finding circuits that are optimized in terms of T-depth [AMMR13, GS13, Sel13, AMM14] and optimal constructions for arbitrary single-qubit unitaries have also been found [KMM13, RS14, Sel15]. These constructions sometimes increase the number of qubits involved by adding ancillas—the use of which can greatly decrease the T-depth of the resulting circuit.

4.2.4 The garden-hose model

The garden-hose model is a combinatorial model of communication complexity, first introduced by Buhrman, Fehr, Schaffner and Speelman [BFSS13]. The recent work by Klauck and Podder [KP14] further investigated the notion, proving several follow-up results. Even though we studied the garden-hose model in Chapter 3, for convenience we will review the basic definitions of the garden-hose model, a few relevant results, and its link to attacks on schemes for position-based quantum cryptography.

Alice has an input $x \in \{0,1\}^n$, Bob has an input $y \in \{0,1\}^n$, and the players want to compute a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ in the following way. Between the two players are s pipes, and, in a manner depending on their respective inputs, the players link up these pipes one-to-one with hoses. Alice also has a water tap, which she can connect to one of these pipes. When $f(x,y) = 0$, the water should exit on Alice's side, and when $f(x,y) = 1$ we want the water to exit at Bob's side. The garden-hose complexity of a function f, written $GH(f)$, then is the least number s of pre-shared pipes the players need to compute the function in this manner.

There is a natural translation from strategies of the garden-hose game to a quantum protocol that routes a qubit to either Alice or Bob depending on their local inputs, up to teleportation corrections. Consider the following quantum task, again dependent on a function f like in the previous paragraph. Alice now receives a quantum state $|\psi\rangle$ and a classical input x, Bob receives input y, and the players are allowed one round of simultaneous communication. If $f(x,y) = 0$, Alice must output $|\psi\rangle$ after this round of communication, and otherwise Bob must output $|\psi\rangle$. We would like to analyze how much pre-shared entanglement the players need to perform this task.

From the garden-hose protocol for f, the players can come up with a strategy for this quantum task that needs at most $GH(f)$ EPR pairs pre-shared. Every pipe corresponds to an EPR pair. If a player's garden-hose strategy dictates a hose between some pipe j and another pipe k, then that player performs a Bell measurement of EPR-halves labeled j and k. Alice's connection of the water tap to a pipe corresponds to a Bell measurement between her input state $|\psi\rangle$ and the local half of an EPR pair. After their measurements, the correct player will hold the state $|\psi\rangle$, up to Pauli corrections incurred by the teleportations. The corrections can be performed after a step of simultaneous communication containing the outcomes of all measurements.

We will describe some of the logic in terms of the garden-hose model, as an abstraction away from the qubits involved. When we refer to a quantum implementation of a garden-hose strategy, we always mean the back-and-forth teleportation as described above.

Equivalence to log-space computation We saw in Chapter 3 that polynomial garden-hose complexity is equivalent to log-space computation – up to a local preprocessing of the inputs. Instead of directly presenting garden-hose protocols, it will sometimes be easier to argue about space-bounded algorithms and then using the equivalence as a black-box translation. We repeat the informal statement of these results here, as a convenient reminder.

Theorem 3.3.15 If $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is log-space computable, then $GH(f)$ is polynomial in n.

Proposition 3.3.17 For a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, if $GH(f)$ is polynomial in n, then f is log-space computable (up to local preprocessing). That is, there exists a polynomial p, functions $\alpha, \beta : \{0,1\}^n \to \{0,1\}^{p(n)}$ and a function $g : \{0,1\}^{p(n)} \times \{0,1\}^{p(n)} \to \{0,1\}$ such that g is log-space computable, and $f(x,y) = g(\alpha(x), \beta(y))$ for all x, y.

The following lemma, proven by Klauck and Podder, will also prove to be useful. Let the number of *spilling pipes* of a garden-hose protocol for a player be the number of possible places the water could possibly exit. More precisely, the number of spilling pipes for Alice for a strategy for a specific input x is the number of different places the water could exit on her side when considering all Bob's possible actions for all possible inputs $y' \in \{0,1\}^n$. The number of spilling pipes for Alice is then the maximum number of spilling pipes over all possible inputs $x' \in \{0,1\}^n$. To be able to chain different parts of a garden-hose protocol together, it is often very convenient to only have a single spilling pipe for each player.

4.2.1. Lemma (Lemma 11 of [KP14]). *A garden-hose protocol P for any function f with multiple spilling pipes can be converted to another garden-hose protocol P' for f that has only one spilling pipe on Alice's side and one spilling pipe on Bob's side. The size of P' is at most 3 times the size of P plus 1.*

Garden-hose protocol for XOR of functions

Klauck and Podder also showed that computing the binary XOR of several protocols is possible with only a linear overhead in total garden-hose complexity [KP14, Theorem 18]. We give an explicit construction for this statement here – the result already follows from the similar construction of [KP14, Lemma 12], except that we obtain a constant which is slightly better than unfolding their (more general) proof.

4.2.2. Lemma. *Let $(f_1, f_2, \ldots, f_k)$ be functions, where each function f_i has garden-hose complexity $GH(f_i)$. Let $c \in \{0,1\}$ be an arbitrary bit. Then,*

$$GH\left(c \oplus \bigoplus_{i=1}^{k} f_i\right) \leq 4 \sum_{i=1}^{k} GH(f_i) + 1 \, .$$

Proof sketch. This statement was proven by Klauck and Podder [KP14, Theorem 18] in a more general form, using the following two steps: First, any garden-hose protocol can be turned into a single-output garden-hose protocol, repeated in this chapter as Lemma 4.2.1, such that the new complexity is at most three times the old complexity. Then, these single-output garden-hose protocols can be used as nodes in a permutation branching program. Our current case is simply an instantiation of that proof for the particular case of the exclusive OR, together with the observation that we can combine both steps into one for this particular case.

For all functions f_i we build a gadget with two input pipes and two output pipes, such that if the water flows in at input pipe labeled $b \in \{0, 1\}$, it flows out at the pipe labeled $f_i \oplus b$. See Figure 4.2.4 for an overview. We use four copies of the garden-hose protocol for f_i.

The open 0 output pipes of the protocol for f_i in copy 0-IN$_i$ are connected to the open 0 output pipes in copy 0-OUT$_i$. The designated source pipe of the original protocol for f_i in copy 0-OUT$_i$ is then guaranteed to be the output.[7] We similarly connect the 1 outputs of 0-IN$_i$ to the 1 outputs of 1-OUT$_i$. This construction, i.e. before adding the 1-IN copy, is exactly the method used to create a single-output protocol. We connect the open 0 pipes of 1-IN$_i$ to the open 0 pipes of 1-OUT$_i$ and the open 1 pipes of the open 1 pipes of 1-IN$_i$ to the open 1 pipes of 0-OUT$_i$.

The gadget then works as claimed by direct inspection. Since all four copies are wired exactly the same, the path of the water through the 'OUT' copy is the reverse of the path it followed through the 'IN' copy, and therefore the water will exit correctly – at the pipe which was the source of the original protocol.

□

4.3 Definition of INQC

An *instantaneous non-local quantum protocol that uses k qubits of entanglement* is a protocol of the following form. We use $\mathcal{S}(A)$ for the set of density matrices on some Hilbert space A.

Alice and Bob start with a fixed, chosen $2k$-qubit state $\eta_{A_e B_e} \in \mathbb{C}^{2^k} \otimes \mathbb{C}^{2^k}$, the entanglement. (Our protocols will all be in the special case where this state is a tensor product of k EPR pairs.) The players receive an input state $\rho \in \mathcal{S}(A_{in} \otimes B_{in})$. Let A_m, A_s, B_m, A_s denote arbitrary-sized quantum registers. Alice applies some quantum operation, i.e. completely positive trace-preserving map, $\mathcal{A}_1 : \mathcal{S}(A_{in} \otimes A_e) \to \mathcal{S}(A_m \otimes A_s)$ and Bob applies the quantum operation $\mathcal{B}_1 : \mathcal{S}(B_{in} \otimes B_e) \to \mathcal{S}(B_m \otimes B_s)$. Alice sends the register A_s to Bob, while simultaneously Bob sends B_s to Alice.

---

[7] This same trick is used in the proof of Lemma 4.2.1 in [KP14, Lemma 11] and in our proof of Lemma 4.5.1.

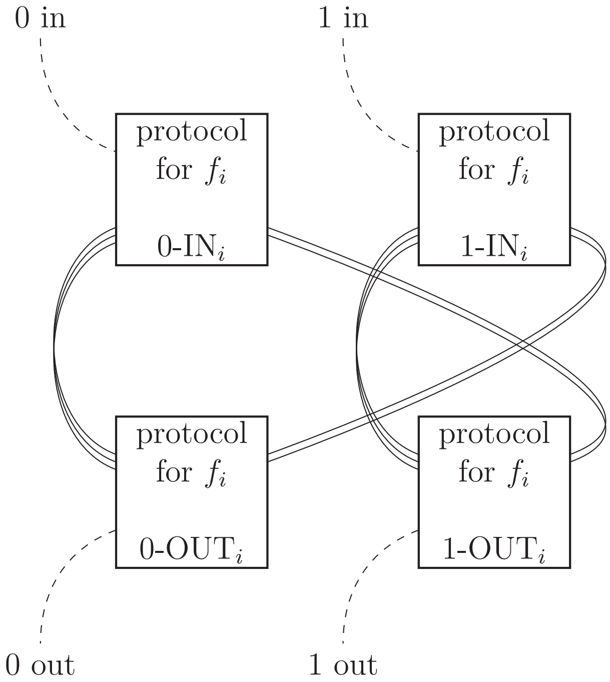
0 in
1 in
protocol for f_i 0-IN$_i$
protocol for f_i 1-IN$_i$
protocol for f_i 0-OUT$_i$
protocol for f_i 1-OUT$_i$
0 out
1 out

Figure 4.1: XOR gadget for any function f_i, total complexity $4GH(f_i)$.

Afterwards Alice applies the quantum operation $\mathcal{A}_2 : \mathcal{S}(A_m \otimes B_s) \to \mathcal{S}(A_{out})$ on her memory and the state she received from Bob, and outputs the result. Likewise Bob applies the operation $\mathcal{B}_2 : \mathcal{S}(B_m \otimes A_s) \to \mathcal{S}(B_{out})$ on the part of the quantum state he kept and outputs the result of this operation.

4.3.1. Definition. Let $\Phi : \mathcal{S}(A_{in} \otimes B_{in}) \to \mathcal{S}(A_{out} \otimes B_{out})$ be a bipartite quantum operation, i.e. a completely positive trace-preserving map, for some input registers A_{in}, B_{in} and output registers A_{out}, B_{out}.

We say that $\mathrm{INQC}_\varepsilon(\Phi)$ is the smallest number k such that there exists an instantaneous non-local quantum protocol that uses k qubits of entanglement, with induced channel $\Psi : \mathcal{S}(A_{in} \otimes B_{in}) \to \mathcal{S}(A_{out} \otimes B_{out})$, so that $\|\Phi - \Psi\|_\diamond \leq \varepsilon$.

For any unitary U, we write $\mathrm{INQC}_\varepsilon(U)$ as a shorthand for $\mathrm{INQC}_\varepsilon(\Phi_U)$, where Φ_U is the induced quantum operation defined by $\rho_{AB} \to U\rho_{AB}U^\dagger$. In this chapter, we assume for simplicity that Alice's and Bob's input and output registers all consist of n qubits.

These definitions are mostly compatible with those given in [BK11], but differ in two ways – both are unimportant for our results in this chapter, but might be relevant for follow-up results, especially when proving lower bounds. Firstly, we made the choice for generality to allow the players to communicate using qubits, instead of just classical messages. As long as the number of communicated qubits is not too large, quantum communication could potentially be replaced by classical communication using teleportation, at the cost of extra entanglement – the counted resource. Secondly, we make the choice to explicitly separate the shared entangled state from the local memory in notation – Beigi and König split the state in a measured and unmeasured part, but do not introduce notation for (free) extra local memory in addition to the shared entangled state.

Whether these choices are reasonable or not will also depend on the exact application. Since we mostly think about applications to position-based quantum cryptography, giving the players, i.e. 'attackers', as much power as possible seems the most natural.

4.4 Low T-count quantum circuits

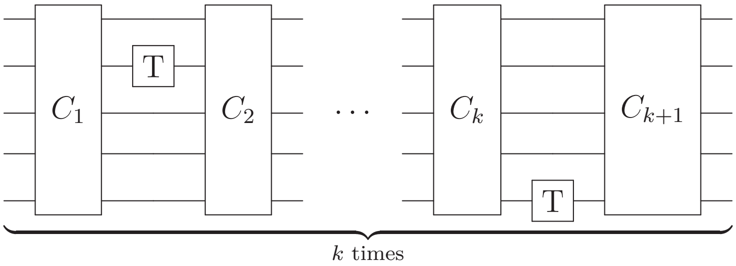

Figure 4.2: A circuit with T-count k. The C_i gates represent subcircuits consisting only of operations from the Clifford group $\mathcal{C}$.

4.4.1. Theorem. *Let C be an n-qubit quantum circuit with gates from the Clifford+T gate set, and let C contain k T-gates in total. Then* $\mathrm{INQC}(C) \leq O(n2^k)$, *i.e., there exists a protocol for two-party instantaneous non-local computation of C which uses a pre-shared entangled state of $O(n2^k)$ EPR pairs.*

Proof. Let Alice's input state be some arbitrary quantum state $|\psi_0\rangle$. We will write the quantum state at step $t \in \{0, \ldots, k\}$, as intermediate result of executing the circuit C for t steps, as $|\psi_t\rangle$. Let C_t be the subcircuit, consisting only of Clifford gates, between the $(t-1)$th and tth T gates. At step t, the circuit alternates between the Clifford subcircuit C_t and a T-gate on some wire w_t which we write as T_{w_t}, i.e.,

$$T_{w_t} = \mathrm{I}^{\otimes w_t - 1} \otimes \mathrm{T} \otimes \mathrm{I}^{\otimes n - w_t - 1} .$$

Because of the nature of the setting, all steps are done instantaneously unless noted otherwise, without waiting for a message of the other party. For example, if the description mentions that one party teleports a qubit, we can instantly describe the qubit as 'being on the other side', but the other party will act on the uncorrected qubit, since the communication will only happen afterwards and simultaneously.

We first give a high-level description of the protocol. Bob teleports his part of the state to Alice, who holds the entire state – up to teleportation corrections. Alice will now apply the first set of Clifford gates, followed by a single T gate. The teleportation corrections (all known to Bob) determine whether the T gate that Alice performs creates an unwanted extra P gate on the state. The extra P gate is created whenever an X correction is present, because of the relation TX = PXT. Therefore, even though Alice holds the state, only Bob knows whether the state has an extra unwanted P gate or not.

To remove the unwanted gate, Alice teleports all n qubits back to Bob, who corrects the phase gate (if present). The players then perform a garden-hose-like trick to keep the form of the key simple, at the cost of doubling the total size at each step.

Now we will give the precise description of the players' actions:

Step 0 Bob performs a Bell measurement to teleport all his $n/2$ qubits to Alice, where we write the needed X-corrections as $b^0_{x,i}$ and Z-corrections $b^0_{z,i}$, for $i = n/2+1, \dots, n$. Now, since the qubits Alice already started with don't need a correction, we have $b^0_{x,i} = b^0_{z,i} = 0$ for $i = 1, \dots, n/2$. Then we write b^0_x and b^0_z for the 0/1 vector containing the X corrections and Z correction respectively. The complete state is $\mathrm{X}^{b^0_x}\mathrm{Z}^{b^0_z}|\psi_0\rangle$, where all qubits are at Alice's side while Bob knows the key.

Step 1.a Alice executes C_1 on the (uncorrected) qubits, so that the state is now

$$C_1\mathrm{X}^{b^0_x}\mathrm{Z}^{b^0_z}|\psi_0\rangle = \mathrm{X}^{\hat{b}^1_x}\mathrm{Z}^{\hat{b}^1_z}C_1|\psi_0\rangle\,,$$

where $(\hat{b}^1_x, \hat{b}^1_z) = f_1(b^0_x, b^0_z)$, with $f_1 : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n \times \mathbb{F}_2^n$ is a formula that consists of relabeling and addition over $\mathbb{F}_2$, and that is known to all parties. I.e. Bob knows all the entries of the vectors $\hat{b}^1_x$ and $\hat{b}^1_z$ that contain the new teleportation corrections.

Step 1.b Alice executes the T gate on the correct wire $w_1 \in \{1, \dots, n\}$ of the uncorrected qubits. Define $\mathbf{b}^1 = \hat{b}^1_{x,w_1}$, the w_1 entry of the vector $\hat{b}^1_x$. The state in Alice's possession is now

$$\begin{aligned}\mathrm{T}_{w_1}\mathrm{X}^{\hat{b}^1_x}\mathrm{Z}^{\hat{b}^1_z}C_1|\psi_0\rangle &= \mathrm{P}_{w_1}^{\mathbf{b}^1}\mathrm{X}^{\hat{b}^1_x}\mathrm{Z}^{\hat{b}^1_z}\mathrm{T}_{w_1}C_1|\psi_0\rangle \\ &= \mathrm{P}_{w_1}^{\mathbf{b}^1}\mathrm{X}^{\hat{b}^1_x}\mathrm{Z}^{\hat{b}^1_z}|\psi_1\rangle\,.\end{aligned}$$

That is, besides the presence of the Pauli gates, depending on the teleportation measurements, the w_1 qubit possibly has an extra phase gate that needs to be corrected before the protocol can continue.

Step 1.c Alice teleports all qubits to Bob, with teleportation outcomes $a_x^1, a_z^1 \in \mathbb{F}_2^n$. We will define the $\mathbf{a}^1$ as the w_1 entry of a_x^1. Bob then has the state

$$\mathrm{X}^{a_x^1}\mathrm{Z}^{a_z^1}\mathrm{P}_{w_1}^{\mathbf{b}^1}\mathrm{X}^{\hat{b}_x^1}\mathrm{Z}^{\hat{b}_z^1}|\psi_1\rangle = \mathrm{P}_{w_1}^{\mathbf{b}^1}\mathrm{X}^{\hat{b}_x^1}\mathrm{Z}^{\hat{b}_z^1}\mathrm{Z}^{\mathbf{a}^1\mathbf{b}^1}\mathrm{X}^{a_x^1}\mathrm{Z}^{a_z^1}|\psi_1\rangle\,.$$

Knowing the relevant variables from his measurement outcomes in the previous steps, Bob performs the operation $\mathrm{X}^{\hat{b}_x^1}\mathrm{Z}^{\hat{b}_z^1}(\mathrm{P}_{w_1}^{\mathbf{b}^1})^\dagger$ to transform the state to

$$\mathrm{Z}^{\mathbf{a}^1\mathbf{b}^1}\mathrm{X}^{a_x^1}\mathrm{Z}^{a_z^1}|\psi_1\rangle\,.$$

Step 1.d For this step the players share two sets of n EPR pairs, one set labeled "$\mathbf{b}^1 = 0$", the other set labeled "$\mathbf{b}^1 = 1$". Bob teleports the state to Alice using the set corresponding to the value of $\mathbf{b}^1$, with teleportation outcomes b_x^2 and b_z^2.

Step 1.e The set of qubits corresponding to the correct value of $\mathbf{b}^1$ are in the state

$$\mathrm{X}^{b_x^2}\mathrm{Z}^{b_z^2}\mathrm{Z}^{\mathbf{a}^1\mathbf{b}^1}\mathrm{X}^{a_x^1}\mathrm{Z}^{a_z^1}|\psi_1\rangle\,.$$

On the set labeled "$\mathbf{b}^1 = 0$", Alice applies $\mathrm{X}^{a_x^1}\mathrm{Z}^{a_z^1}$, and on the set labeled "$\mathbf{b}^1 = 1$" Alice applies $\mathrm{X}^{a_x^1}\mathrm{Z}^{a_z^1}\mathrm{Z}_{w_1}^{\mathbf{a}^1}$, so that the state (at the correct set of qubits) equals

$$\mathrm{X}^{b_x^2}\mathrm{Z}^{b_z^2}|\psi_1\rangle\,.$$

Now this is almost the same situation as before the first step: Alice is in possession of a state for which Bob completely knows the needed teleportation corrections – with the difference that Alice does not know which of the two sets that is.

Steps 2 . . . k The players repeat the protocol from Step 1, but Alice performs all steps in parallel for *all* sets of states. The needed resources then double with each step: two sets for step 2, four for step 3, etc.

Step k+1, final step When having executed this protocol for the entire circuit, Alice only teleports Bob's qubits back to him, i.e. the qubits corresponding to the last $n/2$ wires, instead of the entire state, so that in the correct groups, Alice and Bob are in possession of the state $|\psi_k\rangle$ up to simple teleportation corrections. Then, in their step of simultaneous communication, the players exchange all teleportation measurement outcomes. After receiving these measurement outcomes, the players discard the qubits that did not contain the state, and perform the Pauli corrections on the correct qubits.

The needed EPR pairs for this protocol consist of $n/2$ for Step 0. Then every set uses at most $3n$ pairs: n for the teleportation of Alice to Bob, and $2n$ for the teleportation back. The t-th step of the circuit starts with 2^{t-1} sets of parallel executions, therefore the total entanglement is upper bounded by $n/2 + \sum_{t=1}^{k} 2^{t-1}3n \leq 3n2^k$. $\square$

4.4.1 The Clifford hierarchy

The Clifford hierarchy, also called the Gottesman–Chuang hierarchy, generalizes the definition of the Clifford group of Equation 4.1 in the following way [GC99]. Define $\mathcal{C}_1 = \mathcal{P}$, the first level of the hierarchy, as the Pauli group. Recursively define the k-th level as

$$\mathcal{C}_k = \{U \in U(2^n) \mid \forall \sigma \in \mathcal{P} : U\sigma U^\dagger \in \mathcal{C}_{k-1}\} .$$

Then $\mathcal{C}_2$ is the Clifford group and the next levels consist of increasingly more quantum operations – although for $k \geq 3$ the set $\mathcal{C}_k$ is no longer a group [ZCC08].

The method behind the protocol of Theorem 4.4.1 immediately translates to the related setting of the Clifford hierarchy. Since the scaling of the entanglement cost of our protocol is exponential, both in the number of qubits and in the level of the hierarchy, this observation is mostly interesting when both these quantities are small. We include a proof sketch here. As noted in the introduction, this statement was independently proven by Chakraborty and Leverrier [CL15].

4.4.2. Proposition. *Let U be an n-qubit operation in the k-th level of the Clifford hierarchy, where Alice receives $n/2$ qubits and Bob receives $n/2$ qubits, then* $\mathrm{INQC}(U) \leq O(n4^{nk})$.

Proof sketch. First Bob teleports his qubits to Alice, with n outcomes for X and Z. Alice applies U to the uncorrected state, so that now the state equals $U\mathrm{X}^{b_x}\mathrm{Z}^{b_z}|\psi\rangle = V_{b_x,b_z}U|\psi\rangle$, where V_{b_x,b_z} is an operator in the $(k-1)$-th level of the Clifford hierarchy. Exactly which operator depends on Bob's measurement outcomes b_x, b_z.

Alice teleports the entire state to Bob, with outcomes a_x, a_z, and Bob applies the inverse $V^\dagger_{b_x,b_z}$, so that the state is

$$V^\dagger_{b_x,b_z}\mathrm{X}^{a_x}\mathrm{Z}^{a_z}V_{b_x,b_z}U|\psi\rangle = W_{a_x,a_z,b_x,b_z}U|\psi\rangle ,$$

with W_{a_x,a_z,b_x,b_z} in the $(k-2)$-th level of the Clifford hierarchy. For every possible value of b_x, b_z, the players share a set of n EPR pairs. Bob teleports the state using the set labeled with his measurement outcome b_x, b_z, obtaining teleportation corrections $\hat{b}_x, \hat{b}_z$.

For every set the players repeat this protocol recursively, in the following way. For any set, Alice repeats the protocol as if it were the set used by Bob. At the correct set, Alice effectively knows the values b_x, b_z from the label, and a_x, a_z she knows as own measurement outcomes. The state present is $\mathrm{X}^{\hat{b}_x}\mathrm{Z}^{\hat{b}_z}W_{a_x,a_z,b_x,b_z}U|\psi\rangle$. When Alice applies $W^\dagger_{a_x,a_z,b_x,b_z}$, the state is given by $F_{a_x,a_z,b_x,b_z,\hat{b}_x,\hat{b}_z}U|\psi\rangle$, with F in the $(k-3)$-th level of the Clifford hierarchy. Of this state, effectively only $\hat{b}_x, \hat{b}_z$ is unknown to Alice. Alice teleports this state to Bob using the EPR pairs labeled with a_x, a_z, and the recursive step is complete.

The players continue these steps until the first level of the hierarchy is reached – formed by Pauli operators – after which they can exchange the outcomes of their measurements to undo these and obtain $U|\psi\rangle$.

After t steps, Every teleportation step after the first uses a set of n EPR pairs, picked out of 4^n possibilities corresponding to the Pauli correction of the n qubits teleported in the previous step.

Summing over all rounds gives a total entanglement use of $n\sum_{t=1}^{k} 4^{nt} = O(n4^{nk})$. $\square$

4.5 Conditional application of phase gate using garden-hose protocol

The following lemma connects the difficulty of removing an unwanted phase gate that is applied conditional on a function f, to the garden-hose complexity of f. This lemma is the main technical tool which we use to non-locally compute quantum circuits with a dependence on the T-depth.

4.5.1. LEMMA. *Assume Alice has a single qubit with state* $\mathrm{P}^{f(x,y)}|\psi\rangle$*, for binary strings* $x, y \in \{0,1\}^n$*, where Alice knows the string* x *and Bob knows* y*. Let* $GH(f)$ *be the garden-hose complexity of the function* f*. The following two statements hold:*

1. *There exists an instantaneous protocol without any communication which uses* $2GH(f)$ *pre-shared EPR pairs after which a known qubit of Alice is in the state* $\mathrm{X}^{g(\hat{x},\hat{y})}\mathrm{Y}^{h(\hat{x},\hat{y})}|\psi\rangle$*. Here* $\hat{x}$ *depends only on* x *and the* $2GH(f)$ *bits that describe the measurement outcomes of Alice, and* $\hat{y}$ *depends on* y *and the measurement outcomes of Bob.*
2. *The garden-hose complexities of the functions* g *and* h *are at most linear in the complexity of the function* f*. More precisely,* $GH(g) \leq 4GH(f)+1$ *and* $GH(h) \leq 11GH(f)+2$*.*

Proof. To prove the first statement we will construct a quantum protocol that uses $2GH(f)$ EPR pairs, which is able to remove the conditional phase gate. The quantum protocol uses the garden-hose protocol for f as a black box.

For the second part of the statement of the lemma, we construct garden-hose protocols which are able to compute the teleportation corrections that were incurred by executing our quantum protocol. By explicitly exhibiting these protocols, we give an upper bound to the garden-hose complexity of the X correction g and the Z correction h.

The quantum protocol is shown as Figure 4.3. Alice and Bob execute the garden-hose protocol with the state $\mathrm{P}^{f(x,y)}|\psi\rangle$, i.e. they teleport the state back and forth, with the EPR pairs chosen depending on x and y. Afterwards, if

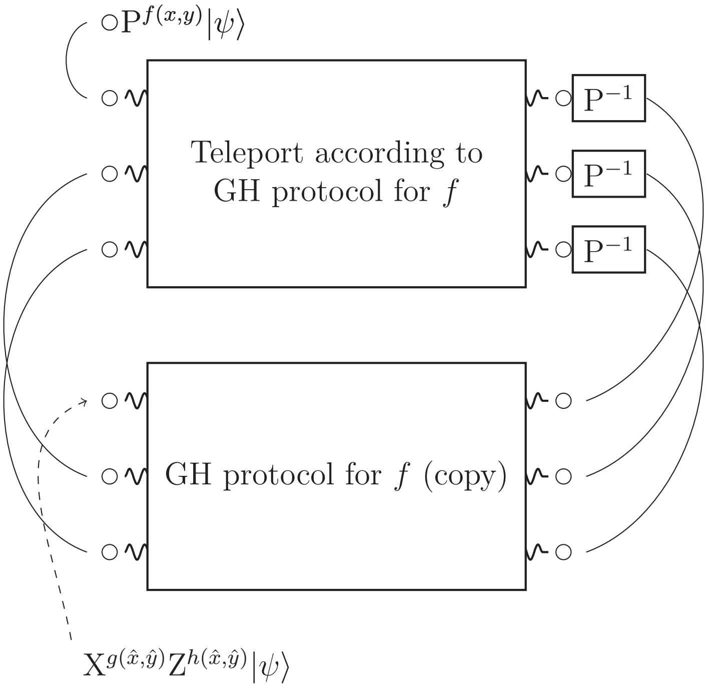
$\mathrm{P}^{f(x,y)}|\psi\rangle$
Teleport according to GH protocol for f
P^{-1}
P^{-1}
P^{-1}
GH protocol for f (copy)
$\mathrm{X}^{g(\hat{x},\hat{y})}\mathrm{Z}^{h(\hat{x},\hat{y})}|\psi\rangle$

Figure 4.3: Schematic overview of the quantum protocol to undo the conditionally-present phase gate on $|\psi\rangle$. The solid connections correspond to Bell measurements.

$f(x, y) = 0$, the qubit will be at one of the unmeasured EPR halves on Alice's side, and if $f(x, y) = 1$ the qubit will be on Bob's side. The state of the qubit will be $\mathrm{X}^{g'(x',y')}\,\mathrm{Z}^{h'(x',y')}\,\mathrm{P}^{f(x,y)}|\psi\rangle = \mathrm{P}^{f(x,y)}\mathrm{X}^{g'(x',y')}\,\mathrm{Z}^{h'(x',y')\oplus f(x,y)g'(x',y')}\,|\psi\rangle$, for some functions g' and h'.

On each qubit on Bob's side, corresponding with an 'open pipe' in the garden-hose model, Bob applies P^{-1}, so that the state of the qubit is now equal to $\mathrm{X}^{g'(x',y')}\,\mathrm{Z}^{h'(x',y')\oplus f(x,y)g'(x',y')}\,|\psi\rangle$. The exact location of our qubit depends on the protocol, and is unknown to both players. Here x' and y' are the measurement outcomes of Alice and Bob in this first half of the protocol.

To return the qubit to a known position without an extra communication step, we employ a trick that uses the reversibility of the garden-hose model. Alice and Bob repeat the exact same garden-hose strategy, except they leave the start open, and connect the open ends between the original and the copy. Alice performs a Bell measurement between the first open qubit in the original, and the first open qubit in the copy, etc. Bob does the same, after he applied the P gates. Afterwards, the qubit will be present in the start location, 'water tap' in garden-hose terminology, of the copied game, since it has followed the exact same path backwards. The final state of the qubit now is $\mathrm{X}^{g(\hat{x},\hat{y})}\,\mathrm{Z}^{h(\hat{x},\hat{y})}\,|\psi\rangle$, for some functions g and h and $\hat{x}$ and $\hat{y}$ the measurement outcomes of Alice and Bob respectively. The total entanglement consumption is $2GH(f)$.

Every measurement corresponds to a connection of two pipes in the garden-hose model, therefore each player performs at most $GH(f)$ teleportation measurements, of which the outcomes can be described by $2GH(f)$ bits.

Label the EPR pairs with numbers from $\{1, 2, \ldots, 2GH(f)\}$, and use the label 0 for the register holding the starting qubit $|\psi\rangle$. Let $\mathcal{A}$ be a list of disjoint pairs of the indices of the EPR pairs that Alice uses for teleportation in this protocol, and let $a_x, a_z \in \{0,1\}^{|\mathcal{A}|}$ be the bit strings that respectively hold the X and Z outcomes of the corresponding Bell measurements. Similarly, let $\mathcal{B}$ be a list of the indices of the EPR pairs that Bob uses, and let $b_x, b_z \in \{0,1\}^{|\mathcal{B}|}$ be the bit strings that hold the measured X and Z corrections.

To show the second part of the statement, we will construct a garden-hose protocol which tracks the newly-incurred Pauli corrections from teleporting the qubit back-and-forth, by following the qubit through the path defined by $\mathcal{A}$ and $\mathcal{B}$.

We will first construct the protocol for the final X-correction, a function we denoted by g. The protocol is also schematically shown as Figure 4.5. Note that to compute the X correction the conditional presence of the phase gate is not important: independent of whether $f(x, y)$ equals 1 or 0, we only need to track the X teleportation corrections that the qubit incurred by being teleported back-and-forth by Alice and Bob. An efficient garden-hose protocol for g is given by the following.

Use two pipes for each EPR pair in the protocol, $2GH(f)$ pairs of 2 pipes each. Label the top pipe of some pair i by I_i, and the bottom pipe by X_i. We will iterate over all elements of $\mathcal{A}$, i.e. all performed Bell measurements by Alice. Consider some element of $\mathcal{A}$, say the k-th pair $\mathcal{A}_k$ which consists of $\{i, j\}$. If the corresponding correction $b_{x,k}$ equals 0, we connect the pipe labeled I_i with the pipe labeled I_j and the pipe labeled X_i with the pipe labeled X_j. Otherwise, if $b_{x,k}$ equals 1, we connect them crosswise, so we connect I_i with X_j and X_i with I_j. Finally, the place where the qubit ends up after the protocol is unique (and is the only unmeasured qubit out of all $2GH(f)$ EPR pairs). For the set of open pipes corresponding to that EPR pair, say number i^*, we use one extra pipe to which we connect X_{i^*}, so that the water ends up at Bob's side for the 1-output. This garden-hose protocol computes the X correction on the qubit, and uses $4GH(f)+1$ pipes in total, therefore $GH(g) \leq 4GH(f) + 1$.

For the Z-correction we can build a garden-hose protocol using the same idea, but there is one complication we have to take care of. At the start of the protocol, there might be an unwanted phase gate present on the state. If some teleportation is performed before this phase gate is corrected, say by Alice with outcomes a_x, a_z, then the effective correction can be written as $\mathrm{X}^{a_x}\mathrm{Z}^{a_z}\mathrm{P} = \mathrm{P}\mathrm{X}^{a_x}\mathrm{Z}^{a_x \oplus a_z}$. That is, for the part of the protocol that the unwanted phase gate is present, a Bell measurement gives a Z-correction whenever the *exclusive or* of the X- and Z-outcomes is 1, instead of just when the Z-outcome is 1. We will therefore use the garden-hose protocol that computes whether $f(x, y) = 1$, that is, compute whether the phase gate is present, and then execute a slightly different garden-hose

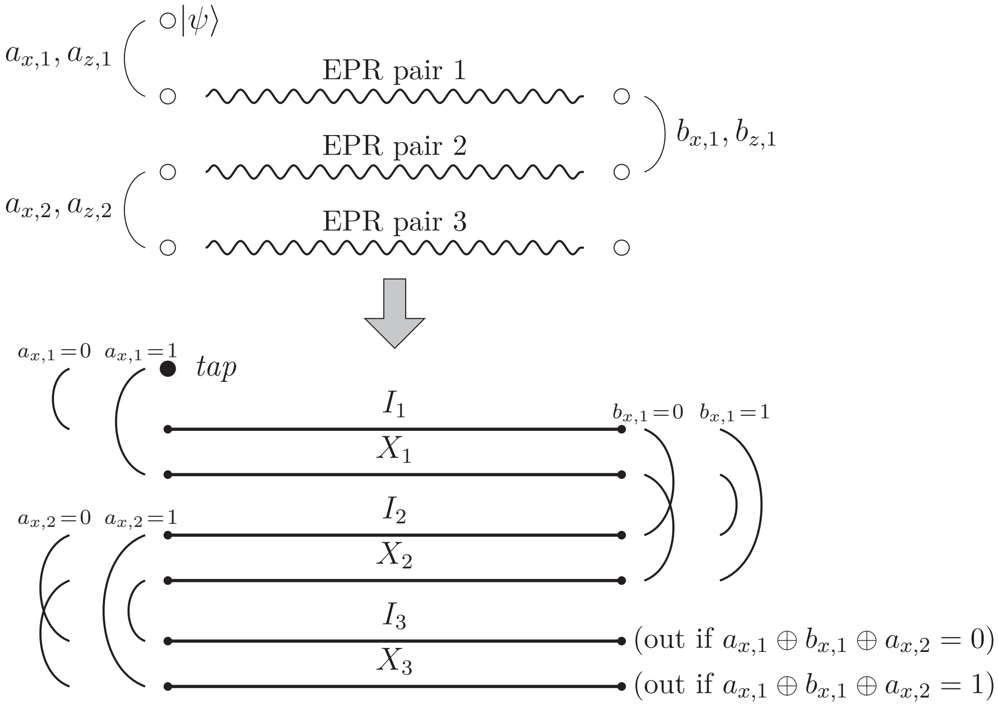

Figure 4.4: Example garden-hose protocol to compute the Pauli X incurred by Alice and Bob teleporting a qubit back-and-forth. When a teleportation requires a Pauli X correction, the corresponding pipes are connected crosswise, and otherwise they are connected in parallel.

protocol for each case.

See Figure 4.5 for an overview of the different parts of this garden-hose protocol for the Z-correction h. Using Lemma 4.2.1 we can transform the garden-hose protocol for f into a garden-hose protocol for f with unique 0 and 1 outputs at Alice's side, of size $3GH(f)$.[8] For the 0 output, i.e. if there was no unwanted phase gate present, we can track the Z corrections in exactly the same way as we did for the X corrections, for a subprotocol of size $4GH(f) + 1$. For the 1 output there was a phase gate present for those teleportations that happened in the protocol before the P^{-1} corrections. For that part of the protocol, we execute the correction-tracking protocol using the XOR of the X- and Z-measurement outcomes. For all teleportations after the phase correction, we again track the correction using just the Z-outcomes, since there is no phase gate present anymore. This part of the garden-hose protocol also uses $4GH(f) + 1$ pipes, for a total of $11GH(f) + 2$.

---

[8] If the unique 0 output has to be at Alice's side, and the unique 1 output at Bob's side, the construction uses $3GH(f) + 1$ pipes. It is easy to show that the construction of Lemma 4.2.1 needs one pipe less if we let Alice have both a designated 0 output and a 1 output.

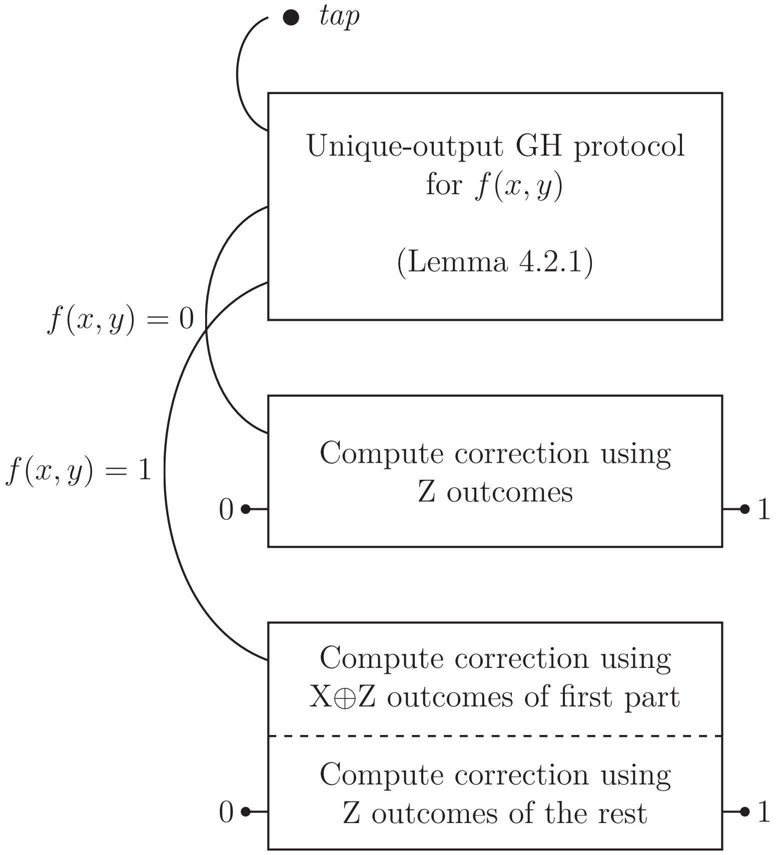
tap
Unique-output GH protocol for $f(x,y)$
(Lemma 4.2.1)
$f(x,y)=0$
$f(x,y)=1$
Compute correction using Z outcomes
0
1
Compute correction using X⊕Z outcomes of first part
Compute correction using Z outcomes of the rest
0
1

Figure 4.5: Sketch of garden-hose protocol for the Z correction. The bottom two boxes use the construction which was used for the X-correction; in the top case using the Z-outcomes for all measurements, in the bottom case using the parity of the X- and Z-outcomes for those teleportations that happened before the unwanted phase gate was removed.

□

4.6 Low T-depth quantum circuits

4.6.1. Theorem. *Let C be an n-qubit T-depth d quantum circuit, then* $\mathrm{INQC}(C) \leq O(\,(68n)^d\,)$. *That is, there exists a protocol for two-party instantaneous non-local computation of C, where each party receives $n/2$ qubits, which uses a pre-shared entangled state of $O(\,(68n)^d\,)$ EPR pairs.*

Proof. As in the proof of Theorem 4.4.1, we write the input state $|\psi\rangle$, and write the correct quantum state after step t of the circuit as $|\psi_t\rangle$. At a step t, the circuit

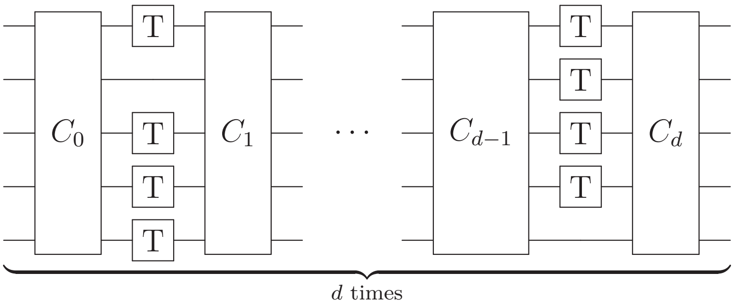

Figure 4.6: An example circuit with T-depth d. The C_i gates represent subcircuits consisting only of operations from the Clifford group $\mathcal{C}$. A layer does not necessarily have a T gate on all wires.

alternates between a *layer* of T gates[9] and a subcircuit consisting of only Clifford gates, C_t.

The high-level idea of this protocol is as follows. During steps 1 to t, Alice will hold the entire uncorrected state and performs a layer of the circuit: she performs a layer of T gates and then a Clifford subcircuit. The Pauli corrections at each step are a function of earlier teleportation outcomes of both Alice and Bob. These functions determine for each qubit whether that qubit now has obtained an unwanted extra P gate when Alice performs the layer of T gates. The players then, for each qubit, correct this extra gate using Lemma 4.5.1 – removing the unwanted phase gate from the qubit in a way that both players still know its location.

At each step we express the corrections as functions of earlier measurements and consider their garden-hose complexity, which is important when using Lemma 4.5.1. The Clifford subcircuit takes the correction functions to the XOR of several earlier functions. We can bound the growth in garden-hose complexity by taking XORs using Lemma 4.2.2. Taken together, the garden-hose complexity grows with a factor of at most a constant times n each step.

We will use $f^t_{x,i}$ to denote the function that describes the presence of an X correction on qubit i, at step t of the protocol. Similarly, $f^t_{z,i}$ is the function that describes the Z correction on qubit i at step t. Both will always be functions of outcomes of earlier teleportation measurements of Alice and Bob. For any t, let m_t be the maximum garden-hose complexity over all the key functions at step t.

Step 0 Bob teleports his qubits, the qubits labeled $n/2$ up to n, to Alice, obtaining the measurement outcomes $b^0_{x,1}, \ldots, b^0_{x,n/2}$ and $b^0_{z,1}, \ldots, b^0_{z,n/2}$. On these uncorrected qubits, Alice executes the Clifford subcircuit C_0.

---

[9] We will assume that for each layer of T gates *all* wires have a T gate. This is only done to avoid introducing extra notation needed when instead the gates are only applied to a subset – the protocol easily generalizes to the more common general situation.

Then, since Bob also knows how C_0 transforms the keys, the functions describing the Pauli corrections can all either be described by a single bit of information which is locally computable by Bob, or are constant and therefore known by both players. Let $f^0_{x,i}$ and $f^0_{z,i}$ be the resulting key function for any qubit i. The garden-hose complexity of all these key functions is constant: $GH(f^0_{x,i}) \leq 3$ and $GH(f^0_{z,i}) \leq 3$, and therefore also for the maximum garden-hose complexity we have $m_0 \leq 3$.

Step $t = 1, \ldots, d$ At the start of the step, the X and Z corrections on any wire i are given by $f^{t-1}_{x,i}$ and $f^{t-1}_{z,i}$ respectively.

Alice applies the T gates on all wires. Any wire i now has an unwanted P if and only if $f^t_{x,i}$ equals 1.

Alice and Bob apply the construction of Lemma 4.5.1, which removes this unwanted phase gate. Let g^t_i be the function describing the extra X correction incurred by this protocol, so that the new X correction can be written as $f^t_{x,i} \oplus g^t_i$. Let h^t_i be the function describing the Z correction, so that the total Z correction is $f^t_{z,i} \oplus h^t_i$. The entanglement cost of this protocol is given by $2GH(f^t_{x,i})$ and the garden-hose complexities of the new functions are at most $GH(g^t_i) \leq 4GH(f^t_{x,i}) + 1$ and $GH(h^t_i) \leq 11GH(f^t_{x,i}) + 2$.

Alice now executes the Clifford subcircuit C_t. The circuit C_t determines how the current Pauli corrections, i.e. the key functions, transform. For a specification of the possible transformations, see Section 4.2.2. For the sake of simplicity, consider new keys that are formed by an XOR of an arbitrary subset of keys that were present in the previous step[10].

Consider the worst case key for our construction: a key which is given by the XOR of all keys that were present when the Clifford subcircuit was executed. A worst-case key function of the form $\bigoplus_{i=1}^n f^{t-1}_{x,i} \oplus g^t_i \oplus f^{t-1}_{z,i} \oplus h^t_i$ then has garden-hose complexity at most

$$\begin{aligned}
m_t &\leq 4\left(\sum_{i=1}^n GH(f^{t-1}_{x,i}) + GH(g^t_i) + GH(f^{t-1}_{z,i}) + GH(h^t_i)\right) + 1 \\
&\leq 4\left(\sum_{i=1}^n GH(f^{t-1}_{x,i}) + 4GH(f^{t-1}_{x,i}) + 1 + GH(f^{t-1}_{z,i}) + 11GH(f^{t-1}_{x,i}) + 2\right) + 1 \\
&\leq 4\left(\sum_{i=1}^n m_{t-1} + 4m_{t-1} + 1 + m_{t-1} + 11m_{t-1} + 2\right) + 1 \\
&= 68nm_{t-1} + 12n + 1\,. \qquad (4.3)
\end{aligned}$$

Step $d+1$**, final step** Alice teleports the last $n/2$ qubits back to Bob. Alice and

---

[10] Do note that the possible key transformations are strictly a subset of these functions – only the transformations generated by the possibilities in Section 4.2.2 are possible.

Bob exchange all results of teleportation measurements and locally perform the needed corrections, using both players' measurement outcomes.

At every step t, the protocol uses at most $2nm_{t-1}$ EPR pairs for the protocol which corrects the phase gate. Using that $m_0 \leq 3$, we can write the upper bound of Equation 4.3 as the closed form $m_t \leq c_1(68n)^t + c_2$, with $c_1 = \frac{216n-2}{68n-1} \approx \frac{54}{17}$ and $c_2 = 3 - \frac{216n-2}{68n-1} \approx -\frac{3}{17}$. The total entanglement use therefore is bounded by

$$\sum_{t=1}^{d} 2nm_{t-1} \leq O(\,(68n)^d\,)\,.$$

□

4.7 Attack on the Interleaved Product protocol

Chakraborty and Leverrier [CL15] recently proposed a scheme for quantum position verification based on the interleaved multiplication of unitaries, the *Interleaved Product protocol*, denoted by $G_{\mathrm{IP}}(n, t, \eta_{\mathrm{err}}, \eta_{\mathrm{loss}})$. The parameter n concerns the number of qubits that are involved in the protocol in parallel, while t scales with the amount of classical information that the protocol uses. Their paper analyzed several different attacks on this scheme, which all required exponential entanglement in the parameter t. In this section, as application of the proof strategy of Theorem 4.6.1, we present an attack on the Interleaved Product protocol which requires entanglement polynomial in t.

The original protocol is described in terms of the actions of hypothetical honest parties and also involves checking of timings at spatial locations. For simplicity, we instead only describe a two-player game, for players Alice and Bob, such that a high probability of winning this game suffices to break the scheme. Let x be a string $x \in_R \{0,1\}^n$, and let U be a random (single-qubit) unitary operation, i.e. a random element of $U(2)$. Alice receives t unitaries $(u_i)_{i=1}^t$, and Bob receives t unitaries $(v_i)_{i=1}^t$ such that $U = \prod_{i=1}^t u_i v_i$. Alice receives the state $U^{\otimes n}|x\rangle$. The players are allowed one round of simultaneous communication. To break the protocol $G_{\mathrm{IP}}(n, t, \eta_{\mathrm{err}}, \eta_{\mathrm{loss}})$, after the round of simultaneous communication the players need to output an identical string $y \in \{\emptyset, 0, 1\}^n$ such that the number of bits where y is different from x is at most $\eta_{\mathrm{err}} n$ and the number of empty results $\emptyset$ is at most $\eta_{\mathrm{loss}} n$. We will consider attacks on the strongest version of the protocol, where we take $\eta_{\mathrm{loss}} = 0$.

For operators A, B, let $\|A\|$ denote the operator norm, and use $\|A - B\|$ as an associated distance measure.

4.7.1. Theorem. *There exists an attack on $G_{\mathrm{IP}}(n, t, \eta_{\mathrm{err}}, \eta_{\mathrm{loss}} = 0)$ that requires $p(t/\eta_{\mathrm{err}})$ EPR pairs per qubit of the protocol, for some polynomial p, and succeeds with high probability.*

Our attack will involve the computation of the unitary $U = \prod_{i=1}^{t} u_i v_i$ in the garden-hose protocol. This is a simple function, but so far we have only defined the garden-hose model for functions with a binary output. Therefore we define an extension of the garden-hose model to functions with a larger output range, where instead of letting the water exit at Alice's or Bob's side, we aim to let the water exit at correctly *labeled pipe*. A short proof of the following proposition is given after the proof of the main theorem.

4.7.2. Proposition. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ be a function, such that f is log-space computable and k is at most $O(\log k)$. Then there exists a garden-hose protocol which uses a polynomial number of pipes, and such that for any input x, y the water exists at Alice's side, at a pipe labeled by the output of $f(x,y)$.*

We will also need a decomposition of arbitrary unitary operations into the Clifford+T gate set. The Solovay–Kitaev theorem is a classic result which shows that any single-qubit quantum gate can be approximated up to precision ε using $O(\log^c(1/\varepsilon))$ gates from a finite gate set, where c is approximately equal to 2. See for example [NC00] for an exposition of the proof. Our constructions use a very particular gate set and we are only concerned with the number of T gates instead of the total number of gates. A recent result by Selinger strengthens the Solovay–Kitaev theorem for this specific case [Sel15][11].

4.7.3. Theorem (Selinger 2015). *Any single-qubit unitary can be approximated, up to any given error threshold $\epsilon > 0$, by a product of Clifford+*T *operators with* T*-count* $11 + 12\log(1/\epsilon)$.

With these auxiliary results in place, we can present our attack on the Interleaved Product protocol.

Proof of Theorem 4.7.1. We will describe the actions taken for any single qubit $U|b\rangle$, with $b \in \{0,1\}$, such that the probability of error is at most ε. The protocol will be attacked by performing these actions on each qubit, n times in parallel. Our construction can be divided in the following four steps.

1. Construct a (polynomial-sized) garden-hose protocol, with a number of pipes s, where the qubit is routed to a pipe labeled with a unitary $\tilde{U}$ which is ε_1-close to the total product U.

2. Decompose the unitaries of all labels in terms of the Clifford+T gate set, using Theorem 4.7.3. In particular, we have a Clifford+T circuit C with T-count $k = O(\log \varepsilon_2)$ such that C is ε_2-close to $\tilde{U}$, and therefore C is at most ε-close to U, where $\varepsilon = \varepsilon_1 + \varepsilon_2$.

---

[11] When the single-qubit unitary is a z-rotation, an even stronger version of the theorem is available [RS14].

3. After executing the garden-hose protocol as a series of teleportations, the state at pipe $\tilde{U}$ can be approximated by $X^{f_x}Z^{f_z}C|\psi\rangle$, with f_x and f_z functions of the connections Alice and Bob made in step 1 and their measurement outcomes. By the construction of Figure 4.5, described in the proof of Lemma 4.5.1, the garden-hose complexities $GH(f_x)$ and $GH(f_z)$ are at most linear in s.

   We can now alternate between applying a single gate of the circuit $C^\dagger$ and using Lemma 4.5.1, k times in total, to obtain a state which only has Pauli corrections left.

4. After Alice measures this final state, she can broadcast the outcome to Bob. Alice and Bob also broadcast their inputs and measurement outcomes, which together determine whether to flip the outcome of Alice's final measurement.

As the first step, we present a log-space computation solving the following problem (equivalent to the input of the protocol, with simplified notation): The input is given by t two-by-two unitary matrices, $u_1, \ldots, u_t$, and we output a matrix $\tilde{U}$ such that $\|\tilde{U} - u_t \ldots u_2 u_1\| \leq \varepsilon_1$, where $\tilde{U}$ is encoded using $O(\log t + \log 1/\varepsilon_1)$ bits. We can then use a simple extension of Theorem 3.3.15 to transform this computation to a garden-hose protocol.

Store the current intermediate outcome of the product in the memory of our computation, using $2\ell + 2$ bits for each entry of the two-by-two matrix, $\ell + 1$ for the real and imaginary part each. Let M_r denote the memory of our log-space computation after r steps, obtained by computing the product $u_r M_{r-1}$ with rounding. Since the rounded matrix entry has a difference of at most $2^{-\ell}$ with the unrounded entry, we can write the precision loss at each step as $M_r = u_r M_{r-1} + \Delta_r$, where Δ_r is some matrix with all entries absolute value at most $2^{-\ell}$. Note that $\|\Delta_r\| \leq 2^{-\ell+1}$.

The total error incurred by the repeated rounding can now be upper bounded by

$$\begin{aligned}
\|M_t - u_t \ldots u_2 u_1\| &\leq \|u_t M_{t-1} + \Delta_t - u_t \ldots u_2 u_1\| \\
&\leq \|\Delta_t\| + \|u_t(M_{t-1} - u_{t-1} \ldots u_2 u_1)\| \\
&\leq 2^{-\ell+1} + \|M_{t-1} - u_{t-1} \ldots u_2 u_1\| \\
&\leq t 2^{-\ell+1}
\end{aligned}$$

Here we use that $\|AB\| \leq \|A\|\|B\|$ together with the unitarity of all u_i. The final step is by iteratively applying the earlier steps t times. If we choose $\ell = \log t + \log 1/\varepsilon_1 + 1$ and note that the final output $\tilde{U}$ is given by M_t, we obtain the bound.

By application of Proposition 4.7.2 we can convert this log-space computation to a garden-hose protocol, using s pipes, where s is polynomial in ε_1 and t. We then teleport the qubit back-and-forth using Bell measurements given by this garden-hose protocol.

As second step, we approximate the unitaries that label each output pipe of the garden-hose protocol of the previous step. In particular, consider the pipe labeled $\tilde{U}$, and say we approximate $\tilde{U}$ using a Clifford+T circuit C. By Theorem 4.7.3, we can write C using $k = 11 + 12\log(1/\varepsilon_2)$ T gates, such that $\|\tilde{U} - C\| \leq \varepsilon_2$. Therefore, defining $\varepsilon = \varepsilon_1 + \varepsilon_2$, we have $\|U - C\| \leq \varepsilon$.

We will perform the next steps for all unmeasured qubits (corresponding to open pipes in the garden-hose model) in parallel. After the simultaneous round of communication, Alice and Bob are then able to pick the correct qubit and ignore the others.

Consider the state of the qubit after the teleportations chosen by the garden-hose protocol. For some functions f_x, f_z, with inputs Alice's and Bob's measurement outcomes, the qubit has state $\mathrm{X}^{f_x}\mathrm{Z}^{f_z}U|b\rangle$. From now on, we will assume this state is exactly equal to $\mathrm{X}^{f_x}\mathrm{Z}^{f_z}C|b\rangle$ – since U is ε-close to C in the operator norm, this assumption adds error probability at most 2ε to the final measurement outcome[12].

Write the inverse of this circuit as alternation between gates from the Clifford group and T gates, $C^\dagger = C_k\mathrm{T}C_{k-1}\mathrm{T}\dots C_1\mathrm{T}C_0$. We will remove C from the qubit by applying these gates, one by one, by repeated application of Lemma 4.5.1. As convenient shorthand, define the state of the qubit after applying the first r layers of $C^\dagger$, i.e. up to and including C_r, of $C^\dagger$ as

$$|\psi_r\rangle = \mathrm{T}^\dagger C_{r+1}^\dagger \mathrm{T}^\dagger C_{r+2} \dots \mathrm{T}^\dagger C_k^\dagger |b\rangle\,.$$

In particular, we have $C_r\mathrm{T}|\psi_{r-1}\rangle = |\psi_r\rangle$.

By exactly the same construction used in the proof of Lemma 4.5.1, shown in Figure 4.5, we observe that the garden-hose complexities of the functions f_x and f_z is at most $2s + 1$. That is, the protocol uses 2 pipes for all of the s EPR pairs, and connects them in parallel if the corresponding X- or Z-correction is 0, or crosswise if the corresponding X- or Z-correction is 1.

We will use divide f_x^r and f_z^r as the functions describing the X and Z corrections at the end of the step r. Define $m_r = \max\{GH(f_x^i), GH(f_z^i)\}$ to be the maximum garden-hose complexity out the of functions describing the X and Z corrections after step r. After Alice executes the Clifford gate C_0, the new key functions f_x^0 and f_z^0 can be written as (the NOT of) an XOR of subsets of the previous keys, e.g., one of the keys could be $f_x \oplus f_z$. By Lemma 4.2.2, we then have that our starting complexities $GH(f_x^0)$ and $GH(f_z^0)$ are at most linear in s.

Now, for any layer $r = 1, 2, \dots, k$: Our qubit starts in the state $\mathrm{X}^{f_x^{r-1}}\mathrm{Z}^{f_z^{r-1}}|\psi_{r-1}\rangle$, for some functions f_x^{r-1}, f_z^{r-1} that each have garden-hose complexity at most m_{r-1}. After Alice performs a T gate, the qubit is in the state

$$\mathrm{T}\mathrm{X}^{f_x^{r-1}}\mathrm{Z}^{f_z^{r-1}}|\psi_{r-1}\rangle = \mathrm{P}^{f_x^{r-1}}\mathrm{X}^{f_x^{r-1}}\mathrm{Z}^{f_z^{r-1}}\mathrm{T}|\psi_{r-1}\rangle\,.$$

---

[12] See for instance [NC00, Box 4.1] for a computation of this added error.

Now, we apply Lemma 4.5.1, costing $2GH(f_x^{r-1})$ EPR pairs, so that Alice has the state

$$\mathrm{X}^{f_x^{r-1}\oplus g_r}\mathrm{Z}^{f_z^{r-1}\oplus h_r}\mathrm{T}|\psi_{r-1}\rangle\,,$$

for some functions g_r and h_r that depend on the measurement results by Alice and Bob. We have that $GH(g_r) \leq 4GH(f_x^{r-1})+1$ and $GH(g_r) \leq 11GH(f_x^{r-1})+2$.

Now Alice applies the Clifford group gate C_r, so that the state becomes

$$C_r\mathrm{X}^{f_x^{r-1}\oplus g_r}\mathrm{Z}^{f_z^{r-1}\oplus h_r}\mathrm{T}|\psi_{r-1}\rangle = \mathrm{X}^{f_x^r}\mathrm{Z}^{f_z^r}|\psi_r\rangle\,.$$

The functions f_x^r and f_z^r can be expressed as XOR of the functions f_x^{r-1}, f_y^{r-1}, g_r, h_r. These functions have garden-hose complexity respectively at most m_{r-1}, m_{r-1}, $4m_{r-1}+1$ and $11m_{r-1}+2$. By application of Lemma 4.2.2, the exclusive OR of these functions therefore at most has garden-hose complexity $m_r \leq 4(m_{r-1} + m_{r-1} + 4m_{r-1} + 1 + 11m_{r-1} + 2) + 1 = 68m_{r-1} + 13$.

Finally, after application of the gates in $\mathcal{C}^\dagger$, Alice has a qubit in a state which is ε-close to $\mathrm{X}^{f_x^r}\mathrm{Z}^{f_z^r}|b\rangle$. Measurement in the computational basis will produce outcome $b \oplus f_x^r$ with high probability. Besides this final measurement, Alice and Bob both broadcast all teleportation measurement outcomes in their step of simultaneous communication. From these outcomes they can each locally compute f_x^r and so derive the bit b from the outcome, which equals $b \oplus f_x^r$, breaking the protocol.

Our total entanglement usage is s for the first step, and then for each of the at most s output pipes, Alice performs the rest of the protocol. For the part of the protocol that undoes the unitary U, we use at most $2\sum_{r=0}^{k-1} m_r$ EPR pairs (for each of the at most s output pipes of the first part). We have $m_0 \leq O(s)$ and $m_r \leq m_0 \cdot 2^{O(k)}$. Since s is polynomial in t and ε_1 and $k = O(\log \varepsilon_2)$, the total protocol uses entanglement polynomial in t and ε. □

Our attack replaces the exponential dependence on t of the attacks presented in [CL15] by a polynomial dependence. For the case of $\eta_{\mathrm{err}} = 0$, we would need an error per qubit of around $\frac{\varepsilon}{n}$ to achieve total error at most ε. In that case, the entanglement required still grows as a polynomial, now with a super-linear dependence of both parameters n and t.

Only the first step of our attack, i.e. the garden-hose protocol which computes a unitary from the inputs of the players, is specific to the interleaved product protocol. This attack can therefore be seen as a blueprint for attacks on a larger class of protocols: any protocol of this same form, where the unitary operation chosen depends on a log-space computable function with classical inputs, can be attacked with entanglement which scales as a polynomial in the size of the classical inputs.

Proof of Proposition 4.7.2. We can split up the computation $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ into k functions that each compute a bit, $f_1, \ldots, f_k$. Since f is a log-space computation, each of these functions is also a log-space computation and therefore has a polynomial-size garden-hose protocol by Theorem 3.3.15. Using Lemma 4.2.1, we can with linear overhead transform each of these protocol into a unique-output protocol, so that the water flows out at a unique pipe when the function is 0 and another unique pipe when the function is 1. Let p be a polynomial so that the single-output garden-hose protocol of each function f_i uses pipes at most $p(n)$.

First use the protocol for f_1, with output pipes labeled 0 and 1. Now each of these output pipes we feed into their own copy of f_2. The 0 output of the first copy we label 00 and its 1 output 10. Similarly, we label the 0 output of the second copy 01 and the 1 output we label 11. By recursively continuing this construction, we build a garden-hose protocol for the function f which uses s pipes, where s is at most

$$s \leq \sum_{i=1}^{k} 2^{i-1} p(n) \leq 2^k p(n) \,.$$

Since we have taken $k = O(\log n)$, this construction uses a number of pipes polynomial in n. □

4.8 Discussion

In this chapter, we combined ideas from the garden-hose model with techniques from quantum cryptography to find a class of quantum circuits for which instantaneous non-local computation is efficient. These constructions can be used as attacks on protocols for quantum position-verification, and could also be translated back into the settings related to physics (most notable the relation between the constraints of relativity theory and quantum measurements) and distributed computing.

The resource usage of instantaneous non-local quantum computation quantifies the non-locality present in a bi- or multi-partite quantum operation, and there is still room for new upper and lower bounds. Any such bounds will result in new insights, both in terms of position-based quantum cryptography, but also in the other mentioned settings.

Some possible approaches for continuing this line of research are as follows:

- Computing the Pauli corrections happens without error in our current construction. Perhaps introducing randomness and a small probability of error – or the usage of entanglement as given in the *quantum garden-hose model* of Section 3.5 – could make this scheme more efficient.

- Future research might be able to extend this type of construction to a wider gate set or model of computation. One could think for example of a Clifford+cyclotomic gate set [FGKM15], match-gate computation [JKMW09], or measurement-based quantum computation [BBD$^+$09, BFK09].

- We presented an attack on the Interleaved Product protocol which required entanglement polynomial in t. Since the exponent of this polynomial was quite large, the scheme could still be secure under realistic assumptions. Since the parameter t concerns the *classical* information that the verifiers send, requiring attackers to manipulate an amount of entanglement which scales linearly with the classical information would already make a scheme unpractical to break in practice – let alone a quadratic or cubic dependence.

- The combination of the garden-hose model with the tool set of blind quantum computation is potentially powerful in other settings.

  For example, following up on Broadbent and Jeffery who published constructions for quantum homomorphic encryption for circuits of low T-gate complexity [BJ15], Dulek, Speelman, and Schaffner [DSS16] developed an improved scheme for quantum homomorphic encryption, based on this combination as presented in (a preprint of the arXiv version of) this chapter.

Chapter 5

Experimental considerations for single-qubit position verification

Current proposals for quantum position-based cryptography require parties to transmit qubits without any loss, at the speed of light. To implement these schemes somewhere on earth, an experimentalist might have to use photons transmitted through optical fibers as carriers of quantum information, encountering two problems.

Firstly, a sizable fraction of photons will be lost in transmission, and secondly, the speed of light in fiber is significantly lower than in a vacuum. We propose an adapted version of the $\mathrm{QPV}_{\mathrm{BB84}}$ protocol and prove its security against a coalition of attackers that do not share prior entanglement. Our security proof uses a reduction to a semidefinite program (SDP), which we solve numerically.

The results in this chapter are based on research performed by the author together with Harry Buhrman, Christian Schaffner and Hugo Zbinden, currently unpublished.

5.1 Introduction

In this chapter we study the following class of protocols for quantum position verification $\mathrm{QPV}_{\mathrm{BB84}}$, first proposed by Kent, Munro, and Spiller [KMS11]. Also see Section 1.1.1 for an introduction to this protocol.

The basic protocol works as follows. Consider two verifiers on a line[1], V_1 and V_2. Verifier V_1 sends a BB84 qubit $|\phi\rangle$ to the prover P and verifier V_2 sends basis information, encoded as binary string, $\theta \in \{+, \times\}$ so that all signals arrive at the prover simultaneously. The symbol '$+$' denotes the computational basis and '$\times$' denotes the Hadamard basis. The prover P is instructed to measure $|\phi\rangle$ in basis θ, getting an outcome b, and broadcast b to all verifiers. Even though the $\mathrm{QPV}_{\mathrm{BB84}}$ protocol is insecure when a coalition of coordinated adversaries share entanglement, this protocol is provably secure against attackers that do not share any entanglement – but performing the protocol does require the honest players to have a perfect experimental setup.

---

[1] For simplicity, we will describe and analyze one-dimensional protocols.

The security of $\mathrm{QPV}_{\mathrm{BB84}}$ against adversaries that do not share entanglement was first proven by Buhrman et al. [BCF$^+$11]. Later work gave security proofs when this scheme is being executed in parallel [TFKW13] and this result was recently improved to show that $\mathrm{QPV}_{\mathrm{BB84}}$ (repeated n times in parallel) is in fact secure whenever the adversaries share less than $n - O(\log n)$ ebits of entanglement[2] [RG15]. This final result is tight, since it is possible to attack the scheme using n EPR pairs.

The creation and manipulation of the quantum states needed for $\mathrm{QPV}_{\mathrm{BB84}}$ is possible with current experimental capabilities and this is a great part of the appeal of studying precisely this scheme. Indeed, at first sight it might seem that experimental setups for the BB84 cryptosystem can be augmented with precise timing to implement the $\mathrm{QPV}_{\mathrm{BB84}}$ protocol (cf. the experimental realization of [LKB$^+$13]). Unfortunately, should we try to actually build an experiment that implements this protocol, we would encounter a few problems that are particular to quantum position verification. The natural adaption to these issues, that first seem only practical in nature, invalidate the security proofs and much care has to be taken to make sure that the schemes do not suddenly become insecure.

To use the scheme to verify position (on earth), the qubits would have to be implemented as photons that travel through fiber. All classical signals can be sent as radio signals that travel at the speed of light.

First, the **speed of light in fiber** is significantly lower than c, the speed of light in vacuum, and therefore the classical messages have to be sent later than the quantum signals, if we want the signals to arrive simultaneously. All the timings will be relative to the *classical* signals – which do travel at light speed. We will show that, even though this adapted timing does not require us to substantially change the protocol, the timing issues make finding a rigorous security proof harder. A more in-depth discussion of the adaptation needed can be found in Section 5.1.3.

The second experimental issue is that of **photon loss**. Even though the accuracy of measurements can be quite good, many of the photons a verifier attempts to send will not be detected by the honest party trying to execute the protocol. We categorize three main sources of loss for this type of experiment: losses in fiber, detector inefficiencies, and the probabilistic nature of a typical photon source, a Poisson distribution parametrized by the mean photon number. We will give a rough estimate for the magnitude of these sources of loss in current state-of-the-art experimental setups, to make our argument more concrete.

---

[2] The lower bound of [RG15] concerns attackers that use only classical communication, besides sharing an entangled state. This is a slightly weaker model than the one of [TFKW13], and the model we currently consider, where the size of the pre-shared entangled state is bounded even when the round of simultaneous communication consists of a quantum message.

Let η be the fraction of photons that are successfully transmitted through the fiber. Under good conditions, we can estimate this loss as 0.2 db per km. This means that for 15 km, we have $\eta \approx 0.5$.

Define p_{inst} to be detector efficiency of an honest party, including the losses incurred by beam splitters, filters and other instruments of the prover's experimental setup. In our feasibility calculation we will estimate p_{inst} to be around 0.5. Let μ be the mean photon number of the pulse. Since the photon count follows a Poisson distribution, this should be tuned so that a probability of a double/triple/etc. photon is very small, since these events can increase the chance a hypothetical adversary breaks the protocol. We will use a value of $\mu = 0.1$ for our current argument.

The probability that a photon, sent by the verifier, arrives at the prover is then given by the *overall detection probability* $p_{det} = \mu \, p_{inst} \, \eta$. This is the fraction that an honest prover can be expected to reply with a measurement result.

Since the mean photon number is less than one, for some of those losses the photon was never sent over fiber, and for those rounds an adversary will also not be able to interact with these photons. We then define the *effective detection probability* $p^*_{det} = p_{inst} \, \eta$ as the probability that the honest prover would be able to answer, whenever a photon is sent (i.e. whenever an attacker has any information to act on)[3].

Extending the original protocol by allowing the prover to answer 'no photon' with large probability completely breaks security by an easy attack. Let Alice and Bob be the two attackers that work together to simulate the actions of an honest prover without being present at the correct location.

The attack on the naively modified scheme goes as follows. For every incoming photon, the attacker Alice measures $|\phi\rangle$ in a basis randomly chosen from $\{+, \times\}$, guessing the value of the basis θ, and sends the basis and result to Bob. After Alice receives the basis information from Bob, and Bob receives the outcome from Alice, they send a reply to their closest verifier with the answer if Alice's guessed basis matches θ, and announce 'no photon' otherwise. It is easy to see that when $1/p^*_{det}$ is at least the number of possible bases, this strategy enables them to reply with the same rate as an honest prover.

We propose a protocol which is still secure under photon losses, and prove its security against malicious adversaries that do not share entanglement before the start of a round. The protocol is a simple extension of the earlier variant, obtained using a larger set of possible bases in which the qubit might be measured. See Figure 5.1 for a description of the protocol.

The proof strategies that were used in earlier analyses of protocols for quantum position-based cryptography do not seem to translate directly to our case, the

---

[3] Of course an adversary could decide to answer even when there was no photon in the fiber to intercept. For these rounds, he will be able to guess the answer with probability at most 0.5. Therefore an easy calculation shows that producing *any* answer on a non-negligible fraction of these rounds will make his chances of fooling the verifiers very small.

0. V_0 picks a random bit $b \in \{0,1\}$, a random basis $\theta \in \{0,\ldots,m-1\}$ and a random number $r \in \{0,\ldots,m-1\}$, and privately sends all three to V_1. The verifier V_0 prepares the state $|b_\theta\rangle$, where we define

$$|0_\theta\rangle := \cos\frac{\pi\theta}{2m}|0\rangle + \sin\frac{\pi\theta}{2m}|1\rangle$$
$$|1_\theta\rangle := \sin\frac{\pi\theta}{2m}|0\rangle - \cos\frac{\pi\theta}{2m}|1\rangle\,.$$

1. V_0 sends the qubit to the location of the prover P. He also sends $m_0 = \theta - r \bmod m$ to P so that it arrives at the same time as the qubit. From the other side V_1 sends $m_1 = r$ to the prover P, so that it arrives at the same time at P as the information from V_0.
2. The prover P calculates $m_0 + m_1 \bmod m = \theta$. P then measures the qubit in the basis given by θ, i.e., the measurement defined by the projectors $\{|0_\theta\rangle\langle 0_\theta|, |1_\theta\rangle\langle 1_\theta|\}$. P sends the outcome b' to both V_0 and V_1. If P has not received a photon because of experimental losses, he replies with ‘no photon’, for which we will use the symbol $\perp$.
3.
    - If one of the verifiers receive a message too late, or if V_0 receives a different message than V_1, the verifiers ABORT. Otherwise:
    - In case the verifiers receive $\perp$ from the prover P, the verifiers register NO PHOTON.
    - If they receive an outcome b which equals b', the verifiers ACCEPT.
    - If b does not equal b', the verifiers note it as WRONG ANSWER.

Figure 5.1: A round of the modified protocol $\mathrm{QPV}_{\text{BB84-e}}$.

inclusion of the ‘no photon’ reply is hard to incorporate. We bypass the difficulties by defining a semidefinite program (SDP) based on the actions of hypothetical attackers and then numerically bounding this SDP. As an intermediate step, we write attacks on the protocol as strategies for a new variant of a *monogamy of entanglement game*, a type of non-local game first introduced by Tomamichel, Fehr, Kaniewski and Wehner [TFKW13], originally used to prove security of parallel repetition of the $\mathrm{QPV}_{\text{BB84}}$ scheme among other things.

The study of non-local quantum correlations through semidefinite programming has been very fruitful and several advanced techniques for giving bounds for the success probability of playing non-local games exist. See, for example, the work of Navascués, Pironio and Acín [NPA08], or semidefinite programming applied to non-local games in [Lan88, Weh06, DLTW08, NV15].

In this chapter, we will use relatively elementary methods of this kind. We directly show that if adversaries can break the protocol with high probability, then there exists a positive semidefinite matrix with various properties. Software for semidefinite optimization can then numerically show that this matrix can not exist and that therefore the protocol is secure.

5.1.1 Results

Using the SDP bounds described in the next section, we can prove limits on how well adversaries can attack a round of the new scheme, for specific experimental parameters. For example we find that with $m = 10$, $p_{err} = 1 \cdot 10^{-3}$, the adversaries can answer with probability at most 0.194, while for the honest prover $p^*_{det} \approx 0.25$. For these experimental parameters the difference could then be detected by repeated rounds, where a dishonest party would eventually have too many errors or declined to answer too many times, compared to the success rate that is achievable by an honest prover.

5.1.2 Related work

Most of this work was done during a visit of Harry Buhrman, Christian Schaffner and Florian Speelman to the University of Geneva in November 2014, and in a follow-up visit of Hugo Zbinden to CWI in Amsterdam in January 2015. Unfortunately, the partial results of this chapter were not made publicly available by the current author at that time – other publications have come out with observations that partially overlap with the work in this chapter.

The independent publication by Qi and Siopsis [QS15], 'Loss-tolerant position-based quantum cryptography'[4], identifies the problem of photon loss in existing single-qubit protocols for quantum position verification and proposes a very similar adapted protocol to be resilient against lost photons. Their work was published before any of our results were available, and they therefore were the first to identify the issue of photon loss for the $\mathrm{QPV}_{\mathrm{BB84}}$ protocol in print.

Though our conclusions are similar, we highlight two differences between this work and [QS15][5].

- The security analysis in [QS15] assumes that attackers always perform a one-dimensional single-qubit projective measurement on the incoming qubit. On the other hand, our proofs are valid for attackers that can perform *any* quantum operation; as long as they shared no entanglement beforehand. The specific SDP formulation we use gives weaker bounds than the results of [QS15], but for more general adversaries.

---

[4] First made available on arXiv in February 2015, and published in Physical Review A in April 2015.

[5] Here we just describe the differences in the security analysis; the work of Qi and Siopsis also reports on their analysis of the continuous-variable analog of this protocol and reflects on the extra attacks possible when the used weak coherent source sends multiple photons.

- The protocol proposed by Qi and Siopsis generalizes the single-qubit $\mathrm{QPV}_{\mathrm{BB84}}$ to pick bases from the entire Bloch sphere, while our work chooses to limit the protocol to the equator which is parametrized by a single angle. We made this choice so that an experimental implementation of the protocol can be performed with fewer optical components, limiting photon losses.

Qi, Lo, Lim, Siopsis, Chitambar, Pooser, Evans, and Grice [QLL$^+$15] also propose a scheme for loss-tolerant quantum position verification. Their results are in the 'free-space' setting, that is, without light-speed limitations. The security proof, like that of Qi and Siopsis, is in a setting where adversaries are not allowed any quantum communication – since we consider a stronger model, we think our results are still of independent interest.

Finally, in very recent work, Johnston, Mittal, Russo, and Watrous [JMRW15] use semidefinite programming to study monogamy of entanglement games, and study an extension of the Navascues–Pironio–Acin hierarchy for these games. They do not directly apply their results to our main question, the study of protocols resilient against loss and light-speed limitations. Combining their new results with the approach we outline in the current chapter could be a possible method to improve the bounds in our security proof.

5.1.3 Security model for limited communication speed

Transmission of qubits as single photons on earth (as opposed to, for example, satellite-to-satellite transmission) is commonly done using optical fiber as medium. Besides the signal loss inherent in fiber, which is the main obstacle we address, the speed of light in the medium is much lower, around two thirds of the speed of light in vacuum c. The security of our protocols critically depends on the signal speed, which determines the possible locations of the quantum information and thus limits the capabilities of the attacker. Even though we only know how to reliably transmit photons at two thirds of light speed with current technology, we should not constrain the attackers in this way, but instead assume that adversaries can transmit any signal with the speed of light – even when forwarding the intercepted qubit which is coming from the verifier.

Contrary to the quantum signals, it is possible to send the *classical* messages reliably and at the speed of light, also for the honest parties. To be able to still argue the security of these protocols, we then must be careful to put constraints on the timing of the classical messages only.

The slowness of the qubit gives the attackers extra time to distribute the qubit amongst themselves; a protocol that sends the complete basis information as a single message would be completely insecure in this setting. Attackers can break such a protocol by using their hypothetical light-speed quantum channel to quickly forward the qubit to the party knowing the basis and so execute a trivial attack.

One natural security model for slow qubits with fast classical information

1. For a random $\theta \in \{0, \ldots, m-1\}$, bit $b \in \{0,1\}$, and a random number $r \in \{0, \ldots, m-1\}$, the verifier V prepares $|\phi\rangle = |b_\theta\rangle$.
2. Alice receives $|\phi\rangle$ and can choose whether to keep it, or let Bob start with $|\phi\rangle$. At this point, the attackers share no entanglement.
3. Alice receives classical message r, Bob receives $\theta - r \bmod m$. Both players simultaneously send one (arbitrarily-sized) quantum message to the other player.
4. Alice and Bob use their received messages and the information they chose to keep, to each send an answer from $\{0, 1, \perp\}$ to the verifier.
5.
   - The verifier outputs ACCEPT if the answers match and are correct.
   - If the answers of Alice and Bob are not the same, the verifier outputs ABORT.
   - In case both Alice and Bob answer $\perp$, the verifier outputs NO PHOTON,
   - If Alice and Bob output the wrong answer, the verifier outputs WRONG ANSWER.

Figure 5.2: A round of attack in the lossy No-PE model for $\text{QPV}_{\text{BB84-e}}$ where the attackers can choose the location of the qubit.

would let the attackers perform a quantum operation Φ before receiving any of the classical messages, but after Alice receives a quantum state ρ_A and Alice and Bob pre-share an auxiliary state $\sigma_{A'B}$. The players then start their attack on the protocol with the state $\Phi(\rho_A \otimes \sigma_{A'B})$, effectively 'distributing the state' beforehand.

Our security proofs concern the no prior-entanglement (No-PE) model, and there is an incompatibility between the slow-qubit security model and not allowing the attackers to share entanglement: they are able to generate new entanglement in their initialization step which distributes the qubit. In keeping with the spirit of the No-PE model, we therefore propose the following adaptation.

Before any round of the protocol starts, the attackers Alice and Bob can route the qubit to any party they desire, and so either Alice or Bob will start with the qubit – security in this model will then still mean that Alice and Bob would need to accurately manipulate entangled states over large distances to break the scheme, while the trivial attack which could happen when Alice and Bob have access to a faster medium than the honest parties is accounted for.

5.1.4 Other protocol modifications

The proposed protocol only picks measurement directions from one circle on the Bloch sphere, a natural extension would pick bases distributed over all one-qubit possibilities. Unfortunately, the gain made by this adjustment is diminished by the extra optical component that would be needed to implement it, which would cause extra losses.

Possible other options, which we do not investigate in-depth in this thesis, include varying the timing of the sent photons, or varying their wavelength. These variations would not make the task of an honest party any harder, but would force an attacker to perform attacks that are significantly more complicated.

5.2 Attack model and proof strategy

We will prove limits on the success probability of an attack on a single round of the protocol via a series of reductions. First, we show that the security of a round of the protocol is implied by a limited success probability for a particular variation of a *monogamy of entanglement game* – a non-local game which was first introduced by Tomamichel, Fehr, Kaniewski and Wehner [TFKW13].

For our case the players can lose the game in different ways, an important difference. A verifier will be much more lenient for ‘no photon’ answers than for wrong answers. The verifier will be very strict in terms of differing answers; if Alice and Bob output a different value, they are instantly caught. Because it is not directly possible to include the ‘no photon’ response in the model of entanglement games, we extend their model. The proof strategy in [TFKW13] will also no longer work for our case, therefore we will use a different method: we define an SDP relaxation, which we numerically solve.

Consider a game with three parties[6]: Alice, Bob, and a verifier V. The game is defined by a collection of measurements of the verifier on his local Hilbert space $\mathcal{H}_V$.

$$\{V_0^\theta, V_1^\theta\}_{\theta \in \{0,\ldots,m-1\}}$$

As in Figure 5.1, we define

$$\begin{aligned} |0_\theta\rangle &:= \cos\frac{\pi\theta}{2m}|0\rangle + \sin\frac{\pi\theta}{2m}|1\rangle \\ |1_\theta\rangle &:= \sin\frac{\pi\theta}{2m}|0\rangle - \cos\frac{\pi\theta}{2m}|1\rangle \,. \end{aligned}$$

Now, let $\mathcal{H}_V = \mathbb{C}^2$ and let the used measurements be the one-dimensional projectors $V_0^\theta = |0_\theta\rangle\langle 0_\theta|$ and $V_1^\theta = |1_\theta\rangle\langle 1_\theta|$.

---

[6] The parties are named differently in the paper that introduced this game - we choose these names for consistency with the other chapter concerning the position-verification setting.

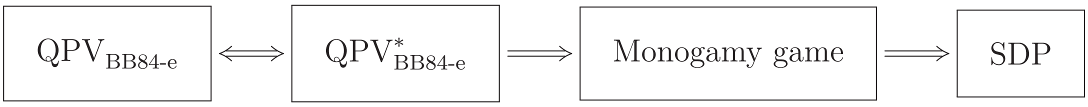

Figure 5.3: A strategy for a round of $\mathrm{QPV}_{\text{BB84-e}}$ can be converted to a strategy for a purified version of the protocol, then into a strategy for the monogamy of entanglement game, and finally to a positive semidefinite matrix with certain properties. Therefore non-existence of this matrix also implies non-existence of the strategy for $\mathrm{QPV}_{\text{BB84-e}}$.

A *strategy* $\mathcal{S}$ to this game is then given by the finite-dimensional Hilbert spaces $\mathcal{H}_A$, $\mathcal{H}_B$, three-outcome projective measurements on these spaces

$$\{A^\theta_0, A^\theta_1, A^\theta_\perp\}_{\theta\in\{0,\ldots,m-1\}}$$

and

$$\{B^\theta_0, B^\theta_1, B^\theta_\perp\}_{\theta\in\{0,\ldots,m-1\}},$$

and a state $|\psi\rangle \in \mathcal{H}_V \otimes \mathcal{H}_A \otimes \mathcal{H}_B$. The assumption that the players use a pure state and that the measurements are projective, instead of general POVM elements, can be made without loss of generality by a standard purification argument (see for example Lemma 9 in [TFKW13]).

For a given strategy, define the probability $p_\perp$ of not playing, i.e. announcing that no photon was received, as

$$p_\perp = \mathop{\mathbb{E}}_{\theta\in\{0,\ldots,m-1\}} \left[\langle\psi| I \otimes A^\theta_\perp \otimes B^\theta_\perp |\psi\rangle\right].$$

Keeping in mind the comparison to the error rate of a photon detector, we define winning probability as the chance of winning *conditional* on playing:

$$p_{win} = \mathop{\mathbb{E}}_{\theta\in\{0,\ldots,m-1\}} \left[\frac{\langle\psi|V^\theta_0 \otimes A^\theta_0 \otimes B^\theta_0|\psi\rangle + \langle\psi|V^\theta_1 \otimes A^\theta_1 \otimes B^\theta_1|\psi\rangle}{\langle\psi|I \otimes A^\theta_0 \otimes B^\theta_0|\psi\rangle + \langle\psi|I \otimes A^\theta_1 \otimes B^\theta_1|\psi\rangle}\right]$$

We will use ε for the maximum probability that the attackers give inconsistent answers. First thought might be to set this to 0, since any single inconsistent answer would let attackers be detected, but if this probability is tiny the attackers might still stay undetected. Instead, we choose to keep it as a (small) numerical parameter, which will have to be picked according to the number of times the protocol is repeated.

5.2.1. PROPOSITION. *If there exists no strategy for the players for the monogamy game that has both conditional success probability p_{win} and 'no play' probability at most $p_\perp$, then there also exists no strategy (in the lossy no prior-entanglement model) for a round of the protocol* $\mathrm{QPV}_{BB84\text{-}e}$ *such that the verifiers output* ACCEPT *with conditional probability at least p_{win} and* NO PHOTON *with probability at most $p_\perp$.*

Proof sketch. First note that without loss of generality, we can assume that Alice does not forward the qubit $|\phi\rangle$, when attacking the protocol $\mathrm{QPV}_{\text{BB84-e}}$ in the way described in Figure 5.1.3; if Alice forwards the qubit to Bob, the attackers have to win a game which is completely equivalent to the game where Alice keeps the qubit. Also in the forwarded case, the players are asked to measure in a basis which comes from the sum of their inputs, therefore a good strategy for a round where Alice forwards can be trivially translated to an equally good strategy where Alice does not forward by swapping the role of the attackers.

Next, note that given previous assumption, we only need to consider the attack strategy for the case where Alice receives 0 and therefore Bob's input completely determines the basis. Indeed, say the attackers have a strategy which is more successful when Alice receives some other classical message m_0', then they could also use this to play the case where $m_0 = 0$ better using that strategy, by using the better strategy with m_1 replaced by $m_1 - m_0' \bmod m$.

Define $\mathrm{QPV}^*_{\text{BB84-e}}$ as a purified version of $\mathrm{QPV}_{\text{BB84-e}}$, where the verifier V_0 prepares an EPR pair and sends one half to the prover (intercepted by attacker Alice), so that they share $\frac{|0\rangle_V|0\rangle_A+|1\rangle_V|1\rangle_A}{\sqrt{2}}$. After randomly selecting $\theta \in \{0,\ldots,m-1\}$, the verifier measures his half in the corresponding basis, getting random outcome b. The reduced state on Alice's side is given by $|b_\theta\rangle$, and therefore security of the purified protocol is completely equivalent to the original variant from the perspective of the attackers.

Finally, a strategy for the purified protocol can be easily transformed into a strategy for the monogamy game, via a similar proof as used in [TFKW13], in the following way. Since Bob only receives classical information, his optimal strategy consists of just forwarding his received bits. Take $|\psi\rangle_{VAB}$ to be the state of the attack after Alice sends her quantum message to Bob. Here register V consists of the qubit of the verifier, which he will measure in basis θ to get the correct outcome b, register A is the quantum memory of Alice and register B contains the message of player Alice to Bob. The measurements $\{A_0^\theta, A_1^\theta, A_\perp^\theta\}_{\theta\in\{0,\ldots,m-1\}}$ and $\{B_0^\theta, B_1^\theta, B_\perp^\theta\}_{\theta\in\{0,\ldots,m-1\}}$ are the measurements Alice and Bob use to obtain their response. □

5.3 Bound by SDP

5.3.1 SDP relaxation of monogamy game

We will show that a good strategy for the modified monogamy of entanglement game implies the existence of a positive semidefinite matrix with certain properties and constraints. We will then use an SDP solver to show that this matrix can not exist, thereby showing security of the protocol.

Assume there exists some strategy $\mathcal{S}$ for the players, with associated projective measurements

$$\{A_0^\theta, A_1^\theta, A_\perp^\theta\}_{\theta\in\{0,\ldots,m-1\}}$$

and

$$\{B_0^\theta, B_1^\theta, B_\perp^\theta\}_{\theta\in\{0,\ldots,m-1\}},$$

and a state $|\psi\rangle \in \mathcal{H}_V \otimes \mathcal{H}_A \otimes \mathcal{H}_B$.

Let G be a $(4m+1)\times(4m+1)$ matrix with real entries. We will think of this matrix as the *Gram matrix* of the set of vectors $|\psi\rangle$, $I \otimes A_a^\theta \otimes I|\psi\rangle$ for $\theta \in \{0,\ldots,m-1\}, a \in \{0,1\}$, and $I\otimes I\otimes B_b^{\theta'}|\psi\rangle$ for $\theta' \in \{0,\ldots,m-1\}, b \in \{0,1\}$, of which there are $4m+1$ total. That is, every entry of the matrix equals the inner product of a corresponding pair of vectors from this set. By deriving bounds on these inner products that would hold for a good strategy, i.e. a strategy corresponding to an undetectable attack on the protocol $\mathrm{QPV}_{\mathrm{BB84\text{-}e}}$, we find constraints on the entries of the matrix G. Note that the optimization only involves the measurement operators of Alice and Bob – the properties of the verifier's measurements (which are known explicitly in our case) will be used to derive constraints on the matrix.

The indices are labeled by symbols, where "I" stands for $|\psi\rangle$, and "A_a^θ" and "$B_b^{\theta'}$" are used for the other vectors respectively. For example, the entry $G(A_a^\theta, B_b^{\theta'})$ is the entry of G that corresponds to the inner product $\langle\psi|(I\otimes I\otimes B_b^{\theta'})(I\otimes A_a^\theta\otimes I)|\psi\rangle = \langle\psi|I\otimes A_a^\theta\otimes B_b^{\theta'}|\psi\rangle$ and the entry $G(A_a^\theta, A_b^{\theta'})$ corresponds to the inner product $\langle\psi|I\otimes A_b^{\theta'}A_a^\theta\otimes I|\psi\rangle$. Even though the latter represents a measurement which is never performed when the game is played, and therefore does not correspond to a probability in the game, the value of that entry can still be seen as inner product between vectors, a hypothetical combining of the strategies for different inputs.

Observe that, for any θ, we can write the third measurement operator as

$$I\otimes A_\perp^\theta\otimes I|\psi\rangle = |\psi\rangle - I\otimes A_0^\theta\otimes I|\psi\rangle - I\otimes A_1^\theta\otimes I|\psi\rangle$$

with similar equalities holding for $B_\perp^{\theta'}$. Therefore we do not need to include the 'no photon' result of the measurements in the matrix, but can write down any constraint that involves the 'no photon' entries as a linear combination of other entries. In our description of the constraints we will still include these, but actual software implementations of the optimization use less memory when we simplify these and optimize over a smaller set of vectors instead.

Since the matrix G is a Gram matrix, it is always positive semidefinite (PSD). Our program will optimize the probability that the players answer something different than $\perp$ over the corresponding entries of all PSD matrices G. The objective function to maximize will then just be the sum

$$\sum_{\substack{a\in\{0,1\}\\ \theta\in\{0,\ldots,m-1\}}} G(A_a^\theta, B_a^\theta)\,.$$

From the form of the vectors we can immediately derive several constraints to impose on G. First of all, since all quantum states are unit vectors, we require that $G(I, I) = 1$. Because all measurements A_a^θ and $B_b^{\theta'}$ are taken to be Hermitian projectors, we have that $G(A_a^\theta, I) = G(A_a^\theta, A_a^\theta)$ and $G(B_b^{\theta'}, I) = G(B_b^{\theta'}, B_b^{\theta'})$ for any θ, a or y, b.

Because of the tensor-product structure between the strategies of the players, we have that $G(A_a^\theta, B_b^{\theta'}) = G(B_b^{\theta'}, A_a^\theta)$ for any θ, a, θ', b.

Since some of the inner products are very common in our derivations, we will abbreviate them. We define

$$p_{a,b}^{\theta,\theta'} = \langle\psi| I \otimes A_a^\theta \otimes B_b^{\theta'} |\psi\rangle$$

corresponding to the case where Alice and Bob receive inputs θ, θ', and produce outputs a, b. We then also define

$$p_a^\theta = p_{a,a}^{\theta,\theta} = \langle\psi| I \otimes A_a^\theta \otimes B_a^\theta |\psi\rangle$$

as a shorthand for the case where both Alice and Bob receive input θ and output a.

The next section we use the extra conditions on winning the game to derive other linear constraints, given as Equations 5.1, 5.2, 5.6 and 5.7.

$$\forall a, b \in \{0, 1, \perp\} \text{ s.t. } a \neq b : \langle\psi| I \otimes A_a^\theta \otimes B_b^\theta |\psi\rangle \leq \varepsilon \qquad \text{(see 5.1)}$$

$$\mathop{\mathbb{E}}_{\theta \in \{0,\ldots,m-1\}} \Big[1 - \langle\psi| I \otimes A_\perp^\theta \otimes B_\perp^\theta |\psi\rangle \Big] \geq p_{det}^* \qquad \text{(see 5.2)}$$

$$p_{a,b}^{\theta,\theta'} \leq \frac{p_{err}(p_0^\theta + p_1^\theta + p_0^{\theta'} + p_1^{\theta'} + 4\varepsilon)}{2 - \|V_a^\theta + V_b^{\theta'}\|} = \begin{cases} \dfrac{p_{err}(p_a^\theta + p_b^{\theta'})}{1 - |\cos\frac{\pi(\theta-\theta')}{2m}|} & \text{if } a = b \\[2ex] \dfrac{p_{err}(p_a^\theta + p_b^{\theta'})}{1 - |\sin\frac{\pi(\theta-\theta')}{2m}|} & \text{if } a \neq b\,. \end{cases} \qquad \text{(see 5.6)}$$

$$\begin{aligned} &p_{0,0}^{\theta,\theta'} + p_{1,0}^{\theta,\theta'} + p_{0,1}^{\theta,\theta'} + p_{1,1}^{\theta,\theta'} \leq \\ &\qquad \left(\frac{1}{1 - |\sin\frac{\pi(\theta-\theta')}{2m}|} + \frac{1}{1 - |\cos\frac{\pi(\theta-\theta')}{2m}|} \right) p_{err}(p_0^\theta + p_1^\theta + p_0^{\theta'} + p_1^{\theta'} + 4\varepsilon) \qquad \text{(see 5.7)} \end{aligned}$$

Now we can fill in these equations, and numerically solve the SDP – for example using the Mosek package. We generated the matrix in Python, with help of the library functions supplied by the Ncpol2sdpa package [Wit15]. We find that with $m = 10$, $p_{err} = 1 \cdot 10^{-3}$, $\varepsilon = 1 \cdot 10^{-7}$ the adversaries can answer with probability at most 0.194, while for the honest prover $p_{det}^* \approx 0.25$. It is possible to choose a low value for ε, since only a single error of that type can be detected by a verifier—also see the comments below.

5.3.2 Deriving the constraints

We will derive some constraints for a strategy that works well (i.e. with high probability) on all inputs θ instead of an average randomly chosen input. This is possible because of the symmetry inherent in the protocol $\mathrm{QPV}_{\mathrm{BB84\text{-}e}}$. Given any attack, where Alice and Bob perform some action on local input θ, we can consider a symmetrized version where Alice and Bob instead perform the local action associated to a randomly chosen input θ', after performing a rotation on their input by $\frac{\pi(\theta'-\theta)}{2m}$. This symmetrized strategy still has the same average-case behavior, but now the worst-case input is exactly equal to the average-case input – the strategy has the same properties for each input θ.

An attack on the protocol that works well for all inputs $\theta \in \{0, \dots, m-1\}$ implies the existence of a strategy for this game with the following properties.

1. If in the original protocol the verifiers V_0 and V_1 receive a different message, any attackers will immediately be detected, since the classical response by an honest player will never differ between V_0 and V_1. Given that the protocol is repeated k times, if the probability per round of a different response is ε, then this causes the attackers to be caught with probability at least $1-(1-\varepsilon)^k$.

   Therefore, a good attack on the protocol will give a strategy for the game with the following property. When players Alice and Bob both receive the same input $\theta \in \{0, \dots, m-1\}$, they will produce a different output with at most the (very low) probability ε.

$$\forall a, b \in \{0, 1, \perp\} \text{ s.t. } a \neq b : \langle\psi| I \otimes A_a^\theta \otimes B_b^\theta |\psi\rangle \leq \varepsilon \tag{5.1}$$

2. The players play with probability at least p_{det}^*.

$$\mathop{\mathbb{E}}_{\theta \in \{0,\dots,m-1\}} \Big[1 - \langle\psi| I \otimes A_\perp^\theta \otimes B_\perp^\theta |\psi\rangle\Big] \geq p_{det}^* \tag{5.2}$$

3. Whenever the players do play, the error probability is low. How can we encode this conditional error probability as constraints on the matrix entries? Out of all times when the verifier gets a 0 as measurement outcome, and the players give a valid answer, this answer is wrong with at most probability given by the measurement error – and the same holds when the verifier measures a 1.

   Define $p_a^\theta = \langle\psi| I \otimes A_a^\theta \otimes B_a^\theta |\psi\rangle$. Then enforcing a low measurement error gives:

$$\frac{\langle\psi| V_0^\theta \otimes A_1^\theta \otimes B_1^\theta |\psi\rangle}{\langle\psi| V_0^\theta \otimes (A_0^\theta + A_1^\theta) \otimes (B_0^\theta + B_1^\theta) |\psi\rangle} \leq p_{err}$$

and

$$\frac{\langle\psi|V_1^\theta \otimes A_0^\theta \otimes B_0^\theta|\psi\rangle}{\langle\psi|V_1^\theta \otimes (A_0^\theta + A_1^\theta) \otimes (B_0^\theta + B_1^\theta)|\psi\rangle} \leq p_{err}$$

therefore, the final constraint on the matrix entry that we derive from the maximum measurement error is given by the sum

$$\begin{aligned}\langle\psi|V_1^\theta \otimes A_0^\theta \otimes B_0^\theta|\psi\rangle + \langle\psi|V_0^\theta \otimes A_1^\theta \otimes B_1^\theta|\psi\rangle &\leq p_{err}\langle\psi|I \otimes (A_0^\theta + A_1^\theta) \otimes (B_0^\theta + B_1^\theta)|\psi\rangle \\ &\leq p_{err}(p_0^\theta + p_1^\theta + 2\varepsilon) \qquad (5.3)\end{aligned}$$

We will now consider the probability that Alice would output a and Bob outputs b should they have received respective different inputs θ and θ', which we call $p_{a,b}^{\theta,\theta'}$. Even though the players always receive the same input when playing a round of the monogamy game, their actions when both receive different inputs are well-defined.

Using the constraints above, we will give a bound on the values of $p_{a,b}^{\theta,\theta'}$.

$$\begin{aligned}2p_{a,b}^{\theta,\theta'} &= 2\langle\psi|I \otimes A_a^\theta \otimes B_b^{\theta'}|\psi\rangle = \langle\psi|(V_a^\theta + V_{1-a}^\theta + V_b^{\theta'} + V_{1-b}^{\theta'}) \otimes A_a^\theta \otimes B_b^{\theta'}|\psi\rangle \\ &= \langle\psi|(V_a^\theta + V_b^{\theta'}) \otimes A_a^\theta \otimes B_b^{\theta'}|\psi\rangle + \langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes B_b^{\theta'}|\psi\rangle \\ &\quad + \langle\psi|V_{1-b}^{\theta'} \otimes A_a^\theta \otimes B_b^{\theta'}|\psi\rangle \\ &\leq \|V_a^\theta + V_b^{\theta'}\| p_{a,b}^{\theta,\theta'} + \langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes I|\psi\rangle + \langle\psi|V_{1-b}^{\theta'} \otimes I \otimes B_b^{\theta'}|\psi\rangle\end{aligned}$$

and therefore

$$p_{a,b}^{\theta,\theta'} \leq \frac{1}{2 - \|V_a^\theta + V_b^{\theta'}\|}\Big(\langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes I|\psi\rangle + \langle\psi|V_{1-b}^{\theta'} \otimes I \otimes B_b^{\theta'}|\psi\rangle\Big). \qquad (5.4)$$

Here the first step used the completeness of the measurement $V_0^\theta + V_1^\theta = I$, for any θ. In the second step we used that for all the measurement operators M, it holds that $M \leq I$.

For $a \in \{0, 1\}$, and any b,θ', we can use the relation in Equation 5.1 to bound

$$\begin{aligned}\langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes I|\psi\rangle &= \langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes (B_a^\theta + B_{1-a}^\theta + B_\perp^\theta)|\psi\rangle \\ &= \langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes B_a^\theta|\psi\rangle + \langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes B_{1-a}^\theta|\psi\rangle \\ &\quad + \langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes B_\perp^\theta|\psi\rangle \\ &\leq \langle\psi|V_{1-a}^\theta \otimes A_a^\theta \otimes B_a^\theta|\psi\rangle + 2\varepsilon, \qquad (5.5)\end{aligned}$$

together with the analogous statement for $\langle\psi|V_{1-b}^{\theta'} \otimes I \otimes B_b^{\theta'}|\psi\rangle$.

Since we have an explicit expression for the measurements of the verifier V, it is not hard to directly compute the operator norm. We prove the following lemma at the end of the chapter.

5.3.1. Lemma. *The operator norm of the sum of two of the measurements of the verifier V is given by*

$$\|V_a^\theta + V_b^{\theta'}\| = \begin{cases} 1 + \left|\cos \dfrac{\pi(\theta-\theta')}{2m}\right| & \textit{if } a = b \\ 1 + \left|\sin \dfrac{\pi(\theta-\theta')}{2m}\right| & \textit{if } a \neq b \end{cases}$$

By combining these expressions, we find a usable bound for sums of the $p_{a,b}^{\theta,\theta'}$ entries, that will form a constraint on our SDP formulation of the problem.

Filling in Equation 5.5 into Equation 5.4, we apply Equation 5.3 to find the bound

$$p_{a,b}^{\theta,\theta'} \leq \frac{p_{err}(p_0^\theta + p_1^\theta + p_0^{\theta'} + p_1^{\theta'} + 4\varepsilon)}{2 - \|V_a^\theta + V_b^{\theta'}\|} = \begin{cases} \dfrac{p_{err}(p_a^\theta + p_b^{\theta'})}{1 - |\cos \frac{\pi(\theta-\theta')}{2m}|} & \text{if } a = b \\ \dfrac{p_{err}(p_a^\theta + p_b^{\theta'})}{1 - |\sin \frac{\pi(\theta-\theta')}{2m}|} & \text{if } a \neq b\,. \end{cases} \tag{5.6}$$

We can also sum four of these probabilities together, to give a slightly better constraint on the sum:

$$p_{0,0}^{\theta,\theta'} + p_{1,0}^{\theta,\theta'} + p_{0,1}^{\theta,\theta'} + p_{1,1}^{\theta,\theta'} \leq \left(\frac{1}{1 - |\sin \frac{\pi(\theta-\theta')}{2m}|} + \frac{1}{1 - |\cos \frac{\pi(\theta-\theta')}{2m}|}\right) p_{err}(p_0^\theta + p_1^\theta + p_0^{\theta'} + p_1^{\theta'} + 4\varepsilon). \tag{5.7}$$

5.3.3 Proof of Lemma 5.3.1

Proof. We compute the norm of the 2×2 matrix $V_a^\theta + V_b^{\theta'}$. For brevity, define $c_\theta = \cos \frac{\pi\theta}{2m}$ and $s_\theta = \sin \frac{\pi\theta}{2m}$, with the analogous definition for $c_{\theta'}$ and $s_{\theta'}$. Then $V_0^\theta = \begin{pmatrix} c_\theta^2 & s_\theta c_\theta \\ s_\theta c_\theta & s_\theta^2 \end{pmatrix}$ and $V_1^\theta = \begin{pmatrix} s_\theta^2 & s_\theta c_\theta \\ s_\theta c_\theta & c_\theta^2 \end{pmatrix}$.

We will compute $\|V_0^\theta + V_0^{\theta'}\|$, the proof for the other three cases can be found similarly.

$$\|V_0^\theta + V_0^{\theta'}\| = \left\|\begin{pmatrix} c_\theta^2 + c_{\theta'}^2 & s_\theta c_\theta + s_{\theta'} c_{\theta'} \\ s_\theta c_\theta + s_{\theta'} c_{\theta'} & s_\theta^2 + s_{\theta'}^2 \end{pmatrix}\right\|$$

This matrix has two eigenvalues, say λ_1 and λ_2. Note that we can express these in terms of the trace $\mathrm{Tr}[V_0^\theta + V_0^{\theta'}] = \lambda_1 + \lambda_2$ and the determinant $\det[V_0^\theta + V_0^{\theta'}] = \lambda_1\lambda_2$. By direct computation we find

$$\mathrm{Tr}[V_0^\theta + V_0^{\theta'}] = c_\theta^2 + s_\theta^2 + c_{\theta'}^2 + s_{\theta'}^2 = 2$$

and

$$\begin{aligned}\det[V_0^\theta + V_0^{\theta'}] &= (c_\theta^2 + c_{\theta'}^2)(s_\theta^2 + s_{\theta'}^2) - (s_\theta c_\theta + s_{\theta'} c_{\theta'})^2\\ &= c_\theta^2 s_{\theta'}^2 + s_\theta^2 c_{\theta'}^2 - 2 c_\theta s_\theta c_{\theta'} s_{\theta'}\\ &= (c_\theta s_{\theta'} - s_\theta c_{\theta'})^2 = \left(\sin\frac{\pi(\theta-\theta')}{2m}\right)^2\\ &= 1 - \left(\cos\frac{\pi(\theta-\theta')}{2m}\right)^2.\end{aligned}$$

Here the last step uses the trigonometric identities $\sin\alpha\cos\beta - \cos\alpha\sin\beta = \sin(\alpha-\beta)$ and $(\sin\alpha)^2 + (\cos\alpha)^2 = 1$. Filling in gives

$$\begin{aligned}\det[V_0^\theta + V_0^{\theta'}] &= \lambda_1\lambda_2 = \lambda_1(2-\lambda_1)\\ &= 1 - (\lambda_1 - 1)^2\\ &= 1 - \left(\cos\frac{\pi(\theta-\theta')}{2m}\right)^2.\end{aligned}$$

Therefore $\lambda_1 = 1 \pm \cos\frac{\pi(\theta-\theta')}{2m}$. Choosing λ_1 to be the largest eigenvalue following convention, we pick the largest option for the sign, depending on the values of θ and θ', giving

$$\lambda_1 = 1 + \left|\cos\frac{\pi(\theta-\theta')}{2m}\right|,$$

and

$$\lambda_2 = 1 - \left|\cos\frac{\pi(\theta-\theta')}{2m}\right|.$$

□

Part II

Catalytic computation

Chapter 6

Catalytic computation

In this chapter we define the notion of a *catalytic-space* computation. This is a computation that has a small amount of clean space available and is equipped with additional auxiliary space, with the caveat that the additional space is initially in an arbitrary, possibly incompressible, state and must be returned to this state when the computation is finished. We show that the extra space (surprisingly) adds extra power to the model: it is possible to compute uniform TC^1-circuits with just a logarithmic amount of clean space. The extra space thus works analogously to a catalyst in a chemical reaction. TC^1-circuits can compute for example the determinant of a matrix, which is not known to be computable in logspace.

In order to obtain our results we study an algebraic model of computation, a variant of straight-line programs. We employ register machines with input registers $x_1, \ldots, x_n$ and work registers $r_1, \ldots, r_m$. The instructions available are of the form $r_i \leftarrow r_i \pm u \times v$, with u, v registers (distinct from r_i) or constants. We wish to compute a function $f(x_1, \ldots, x_n)$ through a sequence of such instructions. The working registers have some arbitrary initial value $r_i = \tau_i$, and they may be altered throughout the computation, but by the end all registers must be returned to their initial value τ_i, except for, say, r_1 which must hold $\tau_1 + f(x_1, \ldots, x_n)$. We show that all of Valiant's class VP, and more, can be computed in this model. This significantly extends the framework and techniques of Ben-Or and Cleve [BC92].

Upper bounding the power of catalytic computation we show that catalytic logspace is contained in ZPP. We further construct an oracle world where catalytic logpace is equal to PSPACE, and show that under the exponential time hypothesis (ETH), SAT can not be computed in catalytic sub-linear space.

The results in this chapter are based on the following publication:

- [BCK$^+$14] Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: Catalytic space. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 857–866, New York, NY, USA, 2014. ACM.

6.1 Introduction

Imagine the following scenario. You want to perform a computation that requires more memory than you currently have available on your computer. One way of dealing with this problem is by installing a new hard drive. As it turns out you have a hard drive but it is full with data, pictures, movies, files, *etc.* You don't need to access that data at the moment but you also don't want to erase it. Can you use the hard drive for your computation, possibly altering its contents temporarily, guaranteeing that when the computation is completed, the hard drive is back in its original state with all the data intact? One natural approach is to compress the data on the hard disk as much as possible, use the freed-up space for your computation and finally uncompress the data, restoring it to its original setting. But suppose that the data is not compressible. In other words, your scheme has to always work no matter the contents of the hard drive. Can you still make good use of this additional space?

In order to study this question we define the following model of computation, which we call *catalytic space*[1]. We equip the standard Turing machine model — which has input, output, and work tapes — with an additional auxiliary tape. We assume that the Turing machine halts on every input and call it *catalytic* if at the end of every computation, the auxiliary tape is unaltered for *every* possible initial setting of its content. As usual in space-bounded computation we limit the amount of work space by a function $s(n)$, usually logarithmic or polynomial. We define the class $\mathsf{CSPACE}(s(n))$ to be the class of sets that are computed by catalytic Turing machines whose work-tape is bounded by $s(n)$ tape cells, and whose auxiliary space is bounded by $2^{s(n)}$ cells.

Intuition tells us that the auxiliary tape is not very useful since its contents must be present in some way at every step of the computation and if these contents are incompressible, effectively no extra space is available. Surprisingly it appears that $\mathsf{CSPACE}(\log n)$, which we call CL, is more powerful than ordinary logspace ($\mathsf{DSPACE}(\log n)$ or L), for we show that $\mathsf{TC}^1 \subseteq \mathsf{CL}$. Note that TC^1 contains NL and even $\#\mathsf{L}$ and other classes that are conjectured to be different from L. We remark that, although the catalytic requirement of the auxiliary space suggests the computation is reversible, it is not sufficient to have just reversibility, since reversible computation schemes [Ben73, LMT97, BTV01] usually require the initial configuration of all the space cells to be set to some fixed initial value, for example all blanks. However, a stronger version of reversibility, that we call *transparent computation*, suffices. Our reversibility framework is related to the work of Ben-Or and Cleve [BC92] but goes beyond it. We show that the techniques of Ben-Or and Cleve stop at the class of problems that are reducible to iterated matrix product (GapL), whereas our model is able to compute TC^1.

---

[1] Catalysis refers to the situation where the rate of a chemical reaction is increased by participation of a substance which is not consumed and is available unaltered after the reaction has taken place.

We don't know what the exact power of catalytic logspace is, but show that it is contained in ZPP – the class of problems which can be solved with randomized algorithms that run in polynomial time on expectation. It could be possible that every problem in P is computable in CL. This would be remarkable. It could be of practical interest in situations when additional clean space is not available, for example when the main memory of a computer is filled with data of an ongoing background computation which may be temporarily stopped, but requires the memory to be unaltered when it continues. On the other hand, CL might be a proper subset of P. There remains the possibility that CL = L. If this is the case then our result implies L = NL, and the intuition that an additional full memory is useless could lead to an approach for proving this collapse. Lastly, we present an oracle relative to which CL = PSPACE, showing the potential, at least in a relativized world, of the auxiliary tape. We also show that under the exponential-time hypothesis [IP99], SAT $\notin$ CSPACE($o(n)$).

6.2 Preliminaries

A short introduction to basic complexity theory can be found in Section 2.4, to put our results into a proper context we also review several problems and related complexity classes here.

L, NL, LOGCFL. By L we denote the class of problems solvable in logspace, by NL the class of problems solvable *non-deterministically* in logspace, and by LOGCFL the class of problems that are logspace many-one reducible to context-free languages. Another equivalent characterization of LOGCFL is as the class of languages accepted by non-deterministic logspace-bounded auxiliary push-down automata (AuxPDAs) running in polynomial time [Sud78].

NC^i, SAC^i, AC^i, TC^i. These are classes of boolean functions computed by polynomial-size circuits of depth $(\log n)^i$. The different classes differ by the set of gates that are allowed in the circuit. NC^i-circuits consist of input gates, constant (0/1) gates, binary (fan-in-2) AND and OR gates, and unary NOT gates. SAC^i-circuits additionally allow for the OR gates to have arbitrary fan-in. AC^i-circuits allow for both AND and OR gates to have arbitrary fan-in. TC^i-circuits are additionally allowed to have MAJ gates of arbitrary fan-in (a MAJ gate decides whether most of its input bits are 1).

GapL, #LOGCFL. We also consider counting classes: GapL is the class of functions obtained by counting the difference between the number of accepting and rejecting paths of a non-deterministic logspace machine; #LOGCFL is the class of functions that count the number of accepting paths of AuxPDAs running in logarithmic space and polynomial time.

VP(R), SkewVP(R). Finally, we will also work with *algebraic* circuits that operate over some ring R. When R is the ring of integers $\mathbb{Z}$, these are also called *arithmetic* circuits. Valiant's class VP(R) [Val79] is the class of (families of) multivariate polynomials over R, computed by algebraic circuits using addition and multiplication gates over R, that have size and degree $n^{O(1)}$ (where n is the number of variables). $\mathsf{SkewVP}(R)$ is the class of multivariate polynomials which can be computed by $\mathsf{VP}(R)$-circuits, with the further restriction that each multiplication gate is binary and such that one of its inputs is either a constant or an input variable.

$\#\mathsf{NC}^i(R)$, $\#\mathsf{SAC}^i(R)$, $\#\mathsf{AC}^i(R)$. These are classes of families of multivariate polynomials over R that are computed by polynomial-size algebraic circuits of depth $(\log n)^i$. Again, these classes differ only by the set of gates that are allowed. $\#\mathsf{NC}^i(R)$-circuits consist of input gates, constant gates (one such gate for each element in R), and binary addition and multiplication gates. $\#\mathsf{SAC}^i(R)$-circuits further allow for the addition gates to have arbitrary fan-in. $\#\mathsf{AC}^i(R)$-circuits can have both addition and multiplication gates of arbitrary fan-in.

Beside circuit families over a ring R that is the same for all input lengths we also consider circuit families where the circuit for inputs of length n computes over a ring R_n, e.g., $\#\mathsf{NC}^1(M_{n^2\times n^2}(\mathbb{Z}))$ consists of families of multivariate polynomials over the ring of integer matrices, where the size of the matrices is n^2 for n being the number of matrix variables.

$\mathsf{DET}_{n,R}$, $\mathsf{IMM}_{n,m,R}$. By $\mathsf{DET}_{n,R}$ we denote the problem of computing a determinant of an $n \times n$ matrix over a ring R. By $\mathsf{IMM}_{n,m,R}$ we denote the problem of computing the product of n matrices, each over the ring R of dimension $m \times m$. We can omit the subscripts when the ring or dimensions are understood from the context. Typically we may think of R being the ring of integers, and $m = n$.

Relationship among these concepts. We now present known relationships among these classes; see Figure 6.1 for an overview.[2]. It is standard knowledge that $\mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{SAC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{TC}^1$, but none of these inclusions is known to be proper. TC^0 is known to contain problems such as computing the sum and the product of n-many n-bit integers, computing the division of two n-bit integers, *etc* [BCH86, RT92, HAM02]. It is also (not-as-well) known that $\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{SAC}^1 = \mathsf{LOGCFL}$ [Ven91].

The complexity of computing the determinant characterizes GapL. More precisely, f is in GapL if and only if it is logspace many-one reducible to $\mathsf{DET}_{n,\mathbb{Z}}$ [Tod91, Dam91, Vin91, Val92]. Cook and others [Coo85, AO94] have shown that the class of problems logspace many-one reducible to DET is the same as the class of problems logspace many-one reducible to IMM.[3] Taken over the integers,

---

[2] Below and in Figure 6.1, inclusion is not meant in a set-theoretic sense, and should be interpreted with the usual caveats that apply to complexity classes; for instance, $\mathsf{NL} \subseteq \mathsf{GapL}$ in the sense that the characteristic function of any NL decision problem is in GapL; or, to give another example, $\#\mathsf{AC}^0(\mathbb{Z}_p) \subseteq \mathsf{TC}^0$ in the sense that any polynomial in $\#\mathsf{AC}^0(\mathbb{Z}_p)$ can be computed in TC^0 using the canonical encoding of $\mathbb{Z}_p$ (see Section 6.4.1) But describing this with full precision would give a cluttered, poorer exposition.

[3] A function f is logspace many-one reducible to the determinant if there is a function g computable in logspace such that $f(x)$ (viewed as a number written in binary) is equal to the

$$\begin{tikzcd}
 & & & & \mathsf{SAC}^1 \arrow[d, equal] & & \\
\mathsf{TC}^0 \arrow[r] \arrow[d, equal] & \mathsf{NC}^1 \arrow[r] & \mathsf{L} \arrow[r] & \mathsf{NL} \arrow[r] \arrow[d] & \mathsf{LOGCFL} \arrow[r] \arrow[d] & \mathsf{AC}^1 \arrow[r] & \mathsf{TC}^1 \arrow[d, equal] \\
\#\mathsf{AC}^0(\mathbb{Z}_{p_n}) \arrow[rrr] & & & \mathsf{GapL} \arrow[r] \arrow[d, equal] & \#\mathsf{LOGCFL} \arrow[rr] \arrow[d, equal] & & \#\mathsf{AC}^1(\mathbb{Z}_{p_n}) \\
 & & & \mathsf{SkewVP}(\mathbb{Z}) \arrow[d, equal] & \mathsf{VP}(\mathbb{Z}) \arrow[d, equal] & & \\
\#\mathsf{AC}^0(\mathbb{Z}_{2^{poly(n)}}) \arrow[uu] & & & \#\mathsf{NC}^1(M_{n\times n}(\mathbb{Z})) & \mathsf{SAC}^1(\mathbb{Z}) & & \#\mathsf{AC}^1(\mathbb{Z}_{2^{poly(n)}}) \arrow[uu]
\end{tikzcd}$$

Figure 6.1: Inclusion diagram for all the classes.

$\mathsf{SkewVP}(\mathbb{Z})$ equals GapL [Tod92], and also $\#\mathsf{NC}^1(M_{n^{O(1)}\times n^{O(1)}}(\mathbb{Z}))$, the class of log-depth fan-in-2 circuits over integer matrices.[4]

$\#\mathsf{SAC}^1(R)$ is actually a characterization of $\mathsf{VP}(R)$ [VSBR83], a deep result of depth-reduction for algebraic circuits. Taken over the integers $\mathsf{VP}(\mathbb{Z})$ equals $\#\mathsf{LOGCFL}$ [Vin91].

The question posed by Valiant [Val79] about the relationship between the determinant and $\mathsf{VP}(\mathbb{Z})$, namely, whether evaluating a $\mathsf{VP}(\mathbb{Z})$ circuit reduces to evaluating the determinant of a matrix that is at most polynomially larger in size (or, equivalently, whether $\mathsf{SkewVP}(\mathbb{Z}) = \mathsf{VP}(\mathbb{Z})$), is no different to the question about the relationship between GapL and $\#\mathsf{LOGCFL}$.

[AAD00, BFS92, RT92] establish a relationship between the classes TC^i and $\#\mathsf{AC}^i(R)$ over various integral rings and finite fields. For instance, it is shown in [RT92] that $\mathsf{TC}^i \subseteq \#\mathsf{AC}^i(\mathbb{Z}_{p(n)})$, where $p(n)$ is any prime number larger than the maximum fan-in of the TC^i circuit to be simulated (for inputs of length n), and that, conversely, $\#\mathsf{AC}^i(\mathbb{Z}_{f(n)}) \subseteq \mathsf{TC}^i$ holds for any function $f(n) = O(2^{poly(n)})$.

A remark on TC^1 versus GapL. Immerman and Landau [IL95] conjecture that computing determinant over the integers is hard for TC^1. However, there is evidence suggesting that this is not the case. Namely, it is known that TC^1 circuits can evaluate $\#\mathsf{AC}^1$ circuits over $\mathbb{Z}_m$, the ring of integers mod m, for exponentially large m.[5] If the Immerman-Landau conjecture were true then $\#\mathsf{SAC}^1$ circuits over the integers — which compute polynomials of degree polynomial in the number of inputs — could simulate TC^1, and hence $\#\mathsf{AC}^1$. But the latter can have super-polynomial degree! This conclusion can not be ruled out entirely, because while polynomials of $n^{O(1)}$ degree over integer variables can not simulate

---

determinant of matrix $g(x)$.

[4] This follows from [BC92, Cle89], see Theorem 6.3.2 below.

[5] This is because TC^0 circuits can evaluate an iterated sum and iterated product of integers, as well as compute the remainder mod m. TC^1 circuits cannot evaluate $\#\mathsf{AC}^1$ circuits over unbounded integers since $\#\mathsf{AC}^1$ circuits represent polynomials of degree up to $n^{O(\log n)}$, and hence the encoding of their output may require super-polynomially many bits.

polynomials of larger degree over integer variables, they could still conceivably simulate polynomials of $n^{\log n}$ degree over integers modulo 2^n ($\mathbb{Z}_{2^n}$). But this does seem unlikely.

6.3 Transparent computation

The model for transparent computation is a variant of straight-line programs. The computational device is a *register machine* equipped with read-write working registers $\vec{r} = r_1, r_2, \ldots, r_m$ and read-only input registers $\vec{x} = x_1, \ldots, x_n$. Each register x_i or r_i holds a value from some designated ring R. The standard set of instructions — called *standard basis* — consists of *instructions* of the form $r_i \leftarrow r_i \pm u \times v$, where u and v are either elements of R ("constants"), or registers different from r_i, and the $+, -$ and $\times$ are the operations of R. These instructions are said to be *reversible*, and for an instruction I, its *inverse* I^{-1} is I with the $+$ or $-$ interchanged.[6] Moreover when at least one of the u and v is an input register or constant we call the instruction *skew*, and the *skew basis* is the standard basis restricted to skew instructions

A *program* for this register machine is a sequence of reversible instructions, and we also call these programs *reversible*. Thus for a reversible program $P = I_1, I_2, \ldots, I_\ell$ we let the *inverse program* P^{-1} be $I_\ell^{-1}, I_{\ell-1}^{-1}, \ldots, I_1^{-1}$. It is easy to verify that P, P^{-1} computes the identity.

We say that a program P *uses register* r if one of its instructions involves this register, e.g., $r_1 \leftarrow r_1 + r_4 \cdot r_7$ uses registers r_1, r_4 and r_7.

We say that $f(\vec{x})$ can be *computed transparently* into a register r_i if there is a reversible program P that when executed on registers $r_1, r_2, \ldots, r_m$ with initial values $\tau_1, \tau_2, \ldots, \tau_m$ ends with value $\tau_i + f(\vec{x})$ in register r_i; the other registers may contain arbitrary values at the end of the computation. (We will always use τ_i to denote the initial value held in register r_i before executing a program.) Clearly, if we have a program that transparently computes f into a register r we can modify it by relabeling registers to compute f transparently into a different register. We may also want P to transparently compute a vector of functions $(f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x}))$ into registers $r_{i_1}, r_{i_2}, \ldots, r_{i_k}$, meaning that the execution of P ends with the value $\tau_{i_j} + f_j(\vec{x})$ in each register r_{i_j}.

Transparent computation is a very special type of reversible computation as it has the additional property that the computation is meaningful regardless of the initial setting of the working registers.[7] Hence the choice of name: the computation is “transparent,” in the sense that it somehow *sees through* the contents of the working registers. This property is *not* universally shared by reversible models of computation. Our model is a variant of the model considered by Coppersmith and Grossman [CG75], and by Ben-Or and Cleve [BC92].

---

[6] Generally speaking, the reversibility property would hold for any instruction of the form $r_i \leftarrow \sigma_{\vec{x},\vec{r}_{\neq i}}(r_i)$, where $\sigma_{\vec{x},\vec{r}_{\neq i}}$ is a permutation of R which may arbitrarily depend on the input registers and on the work registers other than r_i. Also, in principle, different registers could work over different domains. In this chapter we do not make use of these possibilities, but they may appear in future extensions of the model.

[7] Furthermore, and quite remarkably, the following can be shown: let $R(t)$ be the contents of the working registers after executing t instructions of some transparent program, and let

6.3.1. Definition. $\mathsf{TP}(R, s, m)$ is the class of functions transparently computed by reversible programs over the standard basis over ring R, having at most s instructions and using at most m registers. $\mathsf{TP}(R)$ is the class of (families of) functions in $\mathsf{TP}(R, \mathit{poly}, \mathit{poly})$. $\mathsf{SkewTP}(R, s, m)$ and $\mathsf{SkewTP}(R)$ are analogously defined for the skew basis.

Coppersmith and Grossman [CG75] have shown that $\mathsf{TP}(\mathbb{Z}_2, 2^{O(n)}, O(1))$ contains all boolean functions (cf. [Cle89]). The reason why we are interested in transparent computation is because it allows us to restore the work registers to their initial values. For suppose that we have a reversible program P that transparently computes $f(\vec{x})$ into register r_1, while freely modifying the contents of other registers. Then we can take the program P': $r \leftarrow r - r_1, P, r \leftarrow r + r_1, P^{-1}$, where r is a register not used by P. While this new program still transparently computes $f(\vec{x})$ into r, all of the remaining registers are returned to their initial value. We then say that P' *cleanly* (as well as transparently) *computes* $f(\vec{x})$ into register r.

Uniformity. Our class $\mathsf{TP}(R)$ is a non-uniform class. Naturally, we may consider also its uniform variant. All our results in which we simulate circuits by transparent programs essentially preserve the uniformity, so a uniform family of circuits is simulated by a uniform family of transparent programs. There is only a slight loss in our Powering Lemma where we hardwire binomial coefficients into the transparent program. Since the necessary binomial coefficients can easily be computed in logarithmic space the resulting transparent program is still at least logspace uniform if the circuit family is. This also affects all our results that use the Powering Lemma, including our main result on simulation of TC^1. A possible way to avoid this loss in uniformity is to construct very uniform transparent programs that would compute the binomial coefficients.

$X = X_1, \ldots, X_n$ be the input; then for any t, $I(R(t) : X) = I(R(0) : X)$, where I denotes the common information, either in the Shannon or Kolmogorov sense (input and registers must be suitably specified, respectively as a distribution or as a binary string, in order to fit in either framework; details are left to the reader).

In particular, if the initial contents of the registers are independent of the input ($I(R(0) : X) = 0$), then at any point in the computation, the register machine knows nothing about the input, other than whichever specific register X_i it might be accessing directly (as when executing the instruction $r \leftarrow r + X_i$, for instance).

It should be noted, however, that if one is to look at two distinct time-steps t_1 and t_2, some information about X could be derived, i.e., it could hold that $I(R(t_1), R(t_2) : X) > I(R(0) : X)$.

6.3.1 Previous results on this model

It is a natural question to ask: what functions can be transparently computed by small programs over the standard basis, or over other bases? We do not have a precise answer to this question but we will be able to show that all functions in the circuit class TC^1 can be computed transparently by polynomial size programs over the standard basis. This greatly extends the result of Ben-Or and Cleve [BC92] who in essence show that any function in NC^1 can be computed transparently by a polynomial size program using three registers. Cleve in his thesis [Cle89] shows a result slightly stronger than [BC92], namely that iterated matrix product can be computed transparently by polynomial size programs over the standard basis. Indeed, an inspection of the proof, together with the technique of Ben-Or and Cleve, shows that the iterated matrix product can be computed transparently by polynomial size programs over the skew basis. In particular, iterated matrix product of n matrices can be represented by a formula over $R^{m\times m}$ of depth $\log n$. Using the same techniques, we can prove a tight characterization of $\mathsf{SkewTP}(R)$.

6.3.2. Theorem. *Let $f(x_1,\ldots,x_n)$ be a polynomial over a ring R.*

(a) If f can be represented as an entry of a d-depth formula over the ring $M_{m\times m}(R)$, where each entry in each matrix input to this formula is either an element of R, or $\pm x_i$ for some i, and $m = poly(n)$, then f is in $\mathsf{SkewTP}(R, O(m^3 4^d), O(m^2))$.

(b) If f is in $\mathsf{SkewTP}(R, s, m)$, then f can be represented as an entry in the product of s-many $(m+1)\times(m+1)$ such matrices.

Proof. The first part is a restatement of Theorem 3.3.1 of [Cle89]. For the given parameters, it follows that $f \in \mathsf{SkewTP}(R, O(m^3 4^d), O(m^2))$. The only minor difference is that Theorem 3.3.1 of [Cle89] uses standard basis instructions and not our skew basis. However, the inspection of the proof together with the technique of Ben-Or and Cleve [BC92] shows that the theorem is true also for the skew basis.

Now suppose that $f \in \mathsf{SkewTP}(R, S, m)$. Consider the $(m+1)$-dimensional vector $R_0 = (0,\ldots,0,1)$, where the first m entries represent the values of registers $r_1,\ldots,r_n$ used by the program, and the last entry represents a constant one. The skew instruction $r_i \leftarrow r_i \pm r_j \cdot v$, where v is either an element of R or a variable x_i, can be represented by the $(m+1)\times(m+1)$ matrix having 1 on the diagonal, $\pm v$ in the (j,i) position, and 0 elsewhere. The instruction $r_i \leftarrow r_i + v$ can be represented by an identity matrix with the entry $(m+1, i)$ set to v. These matrices will act on the vector R_0 in the same way as their corresponding instructions. If the program transparently computes f into r_1 then the $(1, m+1)$ entry of the product of the matrices corresponding to the program gives f. For each instantiation of $\vec{x}$, this product can be computed by a balanced ($O(\log n)$-depth) tree of product gates over the ring $M_{m\times m}(R)$. □

From the GapL-completeness of IMM over $\mathbb{Z}$, we get:

6.3.3. COROLLARY. $\mathsf{SkewTP}(\mathbb{Z}) = \mathsf{GapL} = \mathsf{SkewVP}(\mathbb{Z}) = \#\mathsf{NC}^1(M_{n^{O(1)} \times n^{O(1)}}(\mathbb{Z}))$.

6.3.2 Getting more

The previous characterization tells us that, to go beyond GapL, we can not restrict ourselves to skew instructions. We will now show how to use reversible programs to transparently compute $\#\mathsf{SAC}^1(R)$. We must then be able to transparently compute binary product and unbounded sum.

6.3.4. LEMMA (BINARY PRODUCT). *Let r_0, r_1, r_2, r_3, r_4 be registers over some ring R. There are reversible programs I_1, I_2, I_3 over the standard basis using registers over R such that for any reversible program P that does not use r_0, r_3 and r_4 and that transparently computes $r_1 \leftarrow \tau_1 + f_1(\vec{x})$ and $r_2 \leftarrow \tau_2 + f_2(\vec{x})$, the program I_1, P, I_2, P^{-1}, I_3 computes $r_0 \leftarrow \tau_0 + f_1(\vec{x}) \times f_2(\vec{x})$. The total length of I_1, I_2, I_3 is eight instructions.*

Proof. The following program computes the required product. The right-hand side indicates the result of applying the instructions on the left-hand side.

```
1. $r_0 \leftarrow r_0 + r_1r_2 + r_1r_4 + r_3r_2$    // $r_0 = \tau_0 + \tau_1\tau_2 + \tau_1\tau_4 + \tau_3\tau_2$

2. $P$                                          // $r_i = \tau_i + f_i(\vec{x})$, for $i = 1, 2$

3. $r_3 \leftarrow r_3 + r_1$                          // $r_3 = \tau_3 + \tau_1 + f_1(\vec{x})$
   $r_4 \leftarrow r_4 + r_2$                          // $r_4 = \tau_4 + \tau_2 + f_2(\vec{x})$
   $r_0 \leftarrow r_0 + r_1r_2$                        // $r_0 = \tau_0 + f_1(\vec{x})f_2(\vec{x})$
                                                //        $+ \tau_1(\tau_4 + \tau_2 + f_2(\vec{x}))$
                                                //        $+ (\tau_3 + \tau_1 + f_1(\vec{x}))\tau_2$

4. $P^{-1}$                                     // $r_i = \tau_i$, for $i = 1, 2$

5. $r_0 \leftarrow r_0 - r_1r_4 - r_3r_2$              // $r_0 = \tau_0 + f_1(\vec{x})f_2(\vec{x})$
```

The first statement, which can be implemented using three standard basis instructions, forms I_1; the statements from line 3 form I_2; and the two instructions corresponding to line 5 form I_3. □

6.3.5. LEMMA (UNBOUNDED SUM). *Let $r_0, r_1, r_2, \ldots, r_k$ be registers over some ring R. There are reversible programs I_1 and I_2 over the standard basis using registers over R such that for any reversible program P that does not use r_0 and that for each $i = 1, \ldots, k$ transparently computes $r_i \leftarrow \tau_i + f_i(\vec{x})$, the program I_1, P, I_2 computes $r_0 \leftarrow \tau_0 + \sum_{i=1}^{k} f_i(\vec{x})$. The total length of I_1, I_2 is $2k$.*

Proof. The following program computes the sum.

1. For each $i = 1, \dots, k$ do $r_0 \leftarrow r_0 - r_i$.
2. P
3. For each $i = 1, \dots, k$ do $r_0 \leftarrow r_0 + r_i$.

The first statement which corresponds to k standard basis instructions forms I_1, and the k instructions from line 3 form I_2. □

6.3.6. COROLLARY. *If R is a ring and f is computed by a depth-d arithmetic circuit with w wires and s gates for binary product and unbounded fan-in addition, then*

$$f \in \mathsf{TP}(R, O(dw2^{d+1}), O(s)) .$$

Proof. Let C be the depth-d circuit for f of given properties. Let us assume that C is layered, that is, each gate at level ℓ takes as its inputs gates at level $\ell - 1$. For every gate g_i of C we will have an auxiliary register r_i into which we will transparently compute the value of g_i. We will compute the values of gates inductively level by level.

If g_i is an input gate then it corresponds either to a constant $c \in R$ or to an input variable x_j. In the former case the instruction $r_i \leftarrow r_i + c$ transparently computes the value of g_i, and in the latter case $r_i \leftarrow r_i + x_j$ does the job. A concatenation of such instructions in arbitrary order for all the input gates gives a program that simultaneously and transparently computes the values of input gates into their associated registers.

Assume that we already have a program $P_{\ell-1}$ that simultaneously and transparently computes the values of gates at the level $\ell - 1$ into appropriate registers. If g_i is a gate at level ℓ then it is either the sum of the values of gates at the level $\ell - 1$ or their binary product. By the Unbounded Sum Lemma or by the Binary Product Lemma, there are programs I_1^i, I_2^i, I_3^i such that $I_1^i, P_{\ell-1}, I_2^i, P_{\ell-1}^{-1}, I_3^i$ transparently computes g_i into r_i. (We can and will assume that I_1^i, I_2^i, I_3^i use different auxiliary registers for different i.) If $g_{i_1}, g_{i_2}, \dots, g_{i_k}$ are the gates at level ℓ then

$$P_\ell = I_1^{i_1}, \dots, I_1^{i_k}, P_{\ell-1}, I_2^{i_1}, \dots, I_2^{i_k}, P_{\ell-1}^{-1}, I_3^{i_1}, \dots, I_3^{i_k}$$

computes simultaneously and transparently the values of the gates at level ℓ into appropriate registers. In this way we obtain a program P_d for transparently computing the value of C.

If the size of the program P_ℓ is S_ℓ then $S_\ell \leq 2S_{\ell-1} + 4w_\ell$, where w_ℓ is the number of wires leading into the gates at the level ℓ. The number of input gates can be bounded by w, so $S_1 \leq w$. Thus $S_\ell \leq 6w2^{\ell-2} \leq w2^{\ell+1}$. Each gate uses at most three registers, and hence our final program will use $O(s)$ registers. This is under the assumption that C is layered. Any circuit can be transformed into a layered one while increasing its number of wires by a factor of at most d. □

We thus get a potentially larger class of functions than that of Ben-Or and Cleve:

6.3.7. Corollary. *For any ring R, $\#\mathsf{SAC}^1(R) \subseteq \mathsf{TP}(R)$. In particular,*

$$\#\mathsf{LOGCFL} = \#\mathsf{SAC}^1(\mathbb{Z}) = \mathsf{VP}(\mathbb{Z}) \subseteq \mathsf{TP}(\mathbb{Z}).$$

6.3.3 Getting TC^1

To go even further and obtain TC^1 we will need the ability to compute the n-th power of a gate. We will show how to do this over commutative rings, but we do not know how to proceed in the non-commutative case.

The following lemma gives a small-length program for computing the iterated product of registers.

6.3.8. Lemma (Iterated product). *There is a program P with $2k+1$ instructions from the standard basis over R that transparently computes, for every $i \leq k$,*

$$r_i \leftarrow \tau_i + m_1 \times \ldots \times m_i,$$

where $m_1, \ldots, m_k$ are either input registers, work registers (different from the r_i), or constants.

Proof. The following program computes the product.

1. For $i = k, \ldots, 2$ do $r_i \leftarrow r_i - r_{i-1} \times m_i$.
2. $r_1 \leftarrow r_1 + m_1$
3. For $i = 2, \ldots, k$ do $r_i \leftarrow r_i + r_{i-1} \times m_i$.

□

Notice that this lemma is different from the binary product or unbounded sum lemmas, in that we do not prove how to inductively compute the iterated product of the outputs of some given program. In fact, we currently do not know how to prove this.

To compute the n-th power over commutative rings, we will need the following variant of the usual binomial expansion.

6.3.9. Lemma. *For any elements a, x of a commutative ring, and any integer $k \geq 1$, the following holds:*

$$(a+x)^k = x^k + \sum_{i=1}^{k} (-1)^{i-1} \binom{k}{i} a^i (a+x)^{k-i}$$

Proof. Let us consider the binomial expansion of $(a+x-a)^k$.

$$\begin{aligned} x^k = (a+x-a)^k &= \sum_{i=0}^{k} \binom{k}{i} (-a)^i (a+x)^{k-i} \\ &= (a+x)^k + \sum_{i=1}^{k} (-1)^i \binom{k}{i} a^i (a+x)^{k-i} \end{aligned}$$

Now the lemma immediately follows. □

6.3.10. Lemma (Powering). *Let k be a positive integer. Let r_0 and r be registers over some* commutative *ring R. There are programs I_1, I_2 and I_3 over the standard basis registers over R such that for any program P that does not use any registers used by I_1, I_2, I_3 other than r and that transparently computes*

$$r \leftarrow \tau + f(\vec{x}),$$

the program I_1, P, I_2, P^{-1}, I_3 computes

$$r_0 \leftarrow \tau_0 + [f(\vec{x})]^k.$$

The total length of I_1, I_2, I_3 is $O(k)$, and $O(k)$ registers are used.

Proof. Assume we have auxiliary registers $r_1, r_2, \ldots, r_k$. Then for the constants $c_i = (-1)^{i-1}\binom{k}{i}$, $i = 1, \ldots, k$, the following program computes the power of $f(\vec{x})$.

1. For $i = 1, \ldots, k$ do $r_0 \leftarrow r_0 + c_i \cdot r_i \cdot r^i$.
2. P
3. For $i = 1, \ldots, k$ do $r_i \leftarrow r_i + r^{k-i}$.
   $r_0 \leftarrow r_0 + r^k$.
4. P^{-1}
5. For $i = 1, \ldots, k$ do $r_0 \leftarrow r_0 - c_i \cdot r_i \cdot r^i$.

This can be seen as before by carefully tracking the contents of the registers, and eventually by applying Lemma 6.3.9. By the Iterated Product Lemma the first line can be implemented using a program over the standard basis of size $O(k)$. This will be I_1. Similarly, line 3 and line 5 can each be implemented by a similar-size program I_2 and I_3, respectively. This would give programs of size $O(k)$ using $O(k)$ registers. □

6.3.11. Lemma (Exact value). *Let p be a prime, R be the field $\mathbb{Z}_p$, and $s \in R$. Let $r_0, r_1, r_2, \ldots, r_k$ be registers over R. There are programs I_1, I_2 and I_3 over the standard basis using registers over R such that for any program P that does not use r_0 and that transparently computes for each $i = 1, \ldots, k$*

$$r_i \leftarrow \tau_i + f_i(\vec{x}),$$

the program I_1, P, I_2, P^{-1}, I_3 computes

$$r_0 \leftarrow \tau_0 + [\![\sum_{i=1}^{k} f_i(\vec{x}) \neq s]\!],$$

where $[\![\sum_{i=1}^{k} f_i(\vec{x}) \neq s]\!]$ equals 1 if $\sum_{i=1}^{k} f_i(\vec{x}) \neq s$ and equals 0 otherwise. The total length of I_1, I_2, I_3 is $O(p + k)$, and $O(p)$ registers are used.

Proof. By the Unbounded Sum Lemma we have programs I_1' and I_2' such that for any program P that simultaneously and transparently computes $r_i \leftarrow r_i + f_i(\vec{x})$, the program $P' = I_1', P, I_2'$ transparently computes $\sum_{i=1}^{k} f_i(\vec{x}) - s$ into an auxiliary register r. The total length of I_1', I_2' is $2k + 1$. Notice, $\sum_{i=1}^{k} f_i(\vec{x}) - s$ is non-zero iff $\sum_{i=1}^{k} f_i(\vec{x}) \neq s$. Since R is a field of size p, by Fermat's little theorem, $(\sum_{i=1}^{k} f_i(\vec{x}) - s)^{p-1}$ is one iff $\sum_{i=1}^{k} f_i(\vec{x}) - s$ is non-zero. Hence, by the Powering Lemma, we have programs I_1'', I_2'', I_3'' such that $I_1'', P', I_2'', P'^{-1}, I_3''$ transparently computes $(\sum_{i=1}^{k} f_i(\vec{x}) - s)^{p-1}$, i.e., $[\![\sum_{i=1}^{k} f_i(\vec{x}) \neq s]\!]$. Setting $I_1 = I_1'', I_1'$, setting $I_2 = I_2', I_2'', (I_2')^{-1}$ and setting $I_3 = (I_1')^{-1}, I_3''$ gives the required programs. Their total length is $2(2k + 1) + O(p)$. □

6.3.12. Corollary. *Let a function f be computed by a depth-d boolean circuit consisting of at most s MAJ-gates, each of fan-in at most k. Let $p > k$ be a prime. Then $f \in \mathsf{TP}(\mathbb{Z}_p, O(dpks4^d), O(dksp))$.*

Proof. First, notice that MAJ gates can be simulated using the Exact Value gates. Indeed, let $b_1, b_2, \ldots, b_k$ be bits where k is even. Then

$$[\![\sum_{j=1}^{k/2} [\![\sum_{i=1}^{k} b_i \neq j]\!] \neq k/2]\!]$$

if and only if

$$\sum_{i=1}^{k} b_i > k/2.$$

Similarly for odd k. Hence, the depth-d circuit C for f consisting of MAJ gates has an equivalent depth-$2d$ circuit C' consisting of the Exact Value gates. The number of gates in C' is at most $O(ks)$. Making C' layered may increase the number of gates by a factor of $2d$. Using the same technique as in the proof of Corollary 6.3.6 we can transparently simulate the computation of C' by a reversible program. Each gate of C' will require additional computation of size $O(k+p)$, and uses $O(p)$ registers. Since, there are $O(dks)$ gates this will contribute by $O(dks(k+p))$ instructions using $O(dksp)$ registers. However, as we proceed layer by layer in constructing the program for C', the number of instructions gets multiplied by a factor of at most 2^{2d} as the instructions for each gate get copied twice at each sub-sequent layer. Hence, in total we obtain a program of length $O(2^{2d}(dk^2s + dkps)) = O(4^d dkps)$. □

Allender and Koucký [AK10] show that for any $\epsilon > 0$, one can simulate MAJ-gate of fan-in n by a uniform constant depth circuit of polynomial size consisting of MAJ-gates of fan-in at most n^ϵ. Hence, in the previous lemma we could use polynomially smaller primes for the cost of increasing the size of the resulting program by a polynomial factor. We can state our main technical result.

6.3.13. Theorem. *For any sequence of primes* $(p_n)_{n\in\mathbb{N}}$ *of size polynomial in* n, $\mathsf{TC}^1 \subseteq \mathsf{TP}(\mathbb{Z}_{p_n})$.

Note, we can find polynomially large primes in logspace so if f is computable by a logspace uniform family of TC^1 circuits then f is transparently computable by a logspace uniform family of polynomial size transparent programs.

Because of the relationship between TC^1 and $\#\mathsf{AC}^1$ the previous theorem allows us to simulate the computation of $\#\mathsf{AC}^1$ circuits over $\mathbb{Z}_m$, the ring of integers modulo m, where m can be exponentially large. Because the degree of the polynomials computed by $\#\mathsf{AC}^1(\mathbb{Z}_m)$ circuits can be as high as $n^{\log n}$, this seems to give a significant improvement over GapL and $\#\mathsf{LOGCFL}$.

6.4 Catalytic computation

A *catalytic Turing machine* is a Turing machine, equipped with a read-only input tape, a work tape[8], and an extra tape — the *auxiliary tape*. For every possible initial setting of the auxiliary tape, at the end of the computation the catalytic Turing machine must have returned the tape to its initial contents.

We say a language L is decided by a catalytic Turing machine M if for any string x, and for any string w representing the initial contents of the auxiliary tape, $M(x,w)$ halts with contents of the auxiliary tape being exactly w and $M(x,w)$ accepts if and only if $x \in L$.

6.4.1. Definition. Let $s, s_a : \mathbb{N} \to \mathbb{N}$. We define the class $\mathsf{CSPACE}(s(n), s_a(n))$ to be the set of all languages that can be decided by a catalytic machine using $O(s(n))$ space of the work tape and $O(s_a(n))$ auxiliary space of the auxiliary tape, for an input of length n.

---

[8] For simplicity, the Turing machine's alphabet is assumed to be $\{0,1\}$, but the model naturally extends to larger alphabets.

As a notational shorthand we define $\mathsf{CSPACE}(s(n)) = \mathsf{CSPACE}(s(n), 2^{O(s(n))})$ as the set of languages that can be decided by a catalytic machine with a work tape of size $s(n)$. We take the auxiliary space exponential in $s(n)$, the largest amount of auxiliary space which can be addressed when using the machine's work tape.

We will pay the most attention to the setting where the machine has work tape of logarithmic size, which we call catalytic logspace or $\mathsf{CL} = \mathsf{CSPACE}(\log n)$.

6.4.1 Simulation of transparent computation by catalytic computation

Our goal is to present now several surprising containments in the catalytic logspace. To achieve that, we will show how to simulate transparent programs in catalytic logspace, how to extract the value of a function from the transparent computation, and how to deal with uniformity issues.

Let us first observe that, in the same way in which one can compose logspace reductions, we can compose constantly many reductions running in catalytic logspace into a single reduction that will also run in catalytic logspace. In this case the total work space will be roughly the sum of the work space used by each of the reductions, but the same auxiliary space can be reused by each of the reductions, since it is returned to its original content after each use. We will heavily use such compositions in this section.

Before proceeding further let us specify what we mean by a uniform sequence of rings.[9] We say that a map $h : R \to \{0,1\}^*$ is a *compact encoding* of the ring R if h is a bijection between R and the lexicographically first $|R|$ strings of length $\ell = \lceil \log_2 |R| \rceil$.[10] We say that a family of rings $(R_n)_{n=1}^{\infty}$ is *logspace uniform*, if there are logspace-bounded Turing machines M, M_+, M_c, M_s and a family $(h_n)_{n=1}^{\infty}$ of compact encodings of $(R_n)_{n=1}^{\infty}$, such that (1) on input $(1^n, h_n(u) \circ h_n(v))$, M outputs $h_n(u \circ v)$, where $u, v \in R$ and $\circ \in \{+, -, \times\}$; (2) with $(1^n, h_n(v))$ written on a read-only tape and $h_n(u)$ written on a read-write tape, M_+ transforms $h_n(u)$ *in-place* into $h_n(u + v)$ for any $u, v \in R_n$ (possibly using $O(\log n)$ of extra space); (3) on input 1^n, M_c outputs $h_n(-1), h_n(0), h_n(1)$ and M_s outputs $|R_n|$.

Examples of logspace uniform families are $(\mathbb{Z}_2)_{n=1}^{\infty}$ and $(\mathbb{Z}_{2^n})_{n=1}^{\infty}$. More generally, if a sequence of numbers $m_1, m_2, \ldots$ is itself logspace uniform in the usual sense then $(\mathbb{Z}_{m_n})_{n=1}^{\infty}$ is logspace uniform. (This follows since addition, multiplication and taking remainder are all computable in logspace, and adding and subtracting two integers can be done *in-place.*) In the case of $\mathbb{Z}_m$, we will make use of the *canonical* compact encoding mapping $n \in \mathbb{Z}_m$ to the n-th $\lceil \log m \rceil$-bit string in the lexicographical order. In this case, the encoding of the binomial coefficients $\binom{n}{k}$ can be computed in $O(\log m)$ space, which will be important for the TC^1 simulation in Section 6.3.3.

---

[9] The well-endowed rings defined by Borodin, Cook and Pippenger [BCP83] are similar, but have different requirements.

[10] The encoding is called compact because in some cases using the lexicographically first $|R|$ strings forces the encoding to be unnatural. This happens in the case of prime fields $\mathbb{F}_{p^n}$ for $p > 2$ and $n > 1$, where the most natural encoding would be n blocks of $\lceil \log_2 p \rceil$ bits, each holding a $\mathbb{Z}_p$ coefficient; but such a natural encoding does not map into the lexicographically first strings of $n\lceil \log_2 p \rceil$ bits, so it is not a *compact* encoding! We will need the encoding to be compact in order to simulate register machines using a full memory.

The following is our key simulation lemma.

6.4.2. LEMMA (CATALYTIC SIMULATION). *For any logspace uniform family of rings* $(R_n)_n$*, there is a logspace catalytic machine* M *that on input* (P, x) *outputs* $f(x)$*, where* P *is a transparent program using registers* $r_1, r_2, \ldots, r_m$ *over* $R_{|x|}$ *that transparently computes* $f(x)$ *into* r_1*. Furthermore,* M *uses* $(m \cdot \lceil \log_2 |R_{|x|}| \rceil)^2$ *bits of auxiliary space, and logarithmic (in terms of length of* P *and* x*) amount of work-space.*

Proof. The machine M will compute $f(x)$ by simulating P in the auxiliary space. Let $n = |x|$. To simulate registers $r_1, \ldots, r_m$ of P the machine will view its auxiliary space as consisting of blocks each having $b = \lceil \log_2 |R_n| \rceil$ bits. Each of the blocks may be used as a register.

Consider first the case when $|R_n|$ is a power of two. Then the first m blocks of the auxiliary space can be used to represent the values of registers $r_1, \ldots, r_m$. As the sequence of rings is uniform, in logspace we can simulate any instruction in the standard basis. Hence, in logspace we can simulate P. To compute the value $f(x)$, we can design a reduction that first outputs the content of r_1, that is the initial content τ_1 of the first block of the auxiliary space, then simulates P and again outputs the content of r_1, this time holding the value $\tau_1 + f(x)$, and finally runs P^{-1} which restores the original content of the auxiliary space. Clearly, this is a reduction running in catalytic logspace. By composing this reduction with one which subtracts the two output values obtained by the previous reduction, we get a program computing $f(x)$.

When $|R_n|$ is not a power of two, we will proceed similarly but we have to represent registers differently. We split our auxiliary space into m groups of mb blocks (each block having b bits as before). Two possibilities may happen: either there is a group in which none of the blocks represents a value from R_n, or each group has a block that represents a value from R_n.

In the first case, if b bits do not represent a value from R_n, then — because our encoding of R_n is compact — they have their first bit set to one. Thus in this case there is a group of mb blocks where the first bit of each block is set to one. These mb bits can be used to simulate m registers of P. We will first erase them, then simulate P, output the content of the first register, which holds $f(x)$, and in the end reset the mb bits back to one.

In the second case, we will use the first block representing a value from R_n in the i-th group to represent the register r_i. Since during the simulation of P, register r_i always contains a value from R_n, it is uniquely determined during the whole computation and we can locate it in logspace. Using the same strategy as in the case of R_n having size of power of two we can compute $f(x)$ while restoring the auxiliary space to its original contents. $\square$

We remark that we could save on the auxiliary space, and instead of using $(m \cdot \lceil \log_2 |R_{|x|}| \rceil)^2$ bits of auxiliary space, we could use only $O(m \cdot \lceil \log_2 |R_{|x|}| \rceil)$ bits if we were to use some stronger compression of the high order bits in the case when there are insufficiently many blocks representing values from R_n.

It is clear that if a sequence of programs $(P_n)_n$ is logspace constructible — where the programs are over some logspace constructible sequence of rings and P_n transparently computes f_n into a register r_1 — then we can compute the function family $(f_n)_n$ in catalytic logspace.

6.4.3. Corollary. *Let $(P_n)_n$ be a logspace uniform sequence of programs over some logspace constructible sequence of rings. Let P_n transparently compute f_n into a register r_1. Then the function family $(f_n)_n$ is in catalytic logspace.*

We remark that our constructions of transparent programs in Section 6.3 are all logspace-uniform. Thus, from the results in Section 6.3 we conclude, quite surprisingly, that a computer which has plenty of occupied memory is (to the extent we believe that $\mathsf{TC}^1 \not\subseteq \mathsf{L}$) more powerful than one that does not.

6.4.4. Theorem. $\mathsf{TC}^1 \subseteq \mathsf{CL}$*, for logspace uniform TC^1.*

The Ben-Or & Cleve construction of Theorem 6.3.2(a) is also uniform. From this (using Chinese remaindering computable in logspace) we obtain a result incomparable to the above:

6.4.5. Theorem. *Iterated matrix product of n matrices over $\mathbb{Z}$, each of dimension $m(n) \times m(n)$, can be computed in logspace with $O(m(n)^2 \cdot \log n)$ bits of auxiliary space. In particular, the iterated matrix product of n matrices over $\mathbb{Z}$, each of dimension $2^{\sqrt{\log n}} \times 2^{\sqrt{\log n}}$, can be computed in logspace with sub-polynomial ($2^{O(\sqrt{\log n})}$) auxiliary space.*

Thus even if the auxiliary space is of less than polynomial size, in catalytic logspace we can still compute functions that are not known to be in the ordinary logspace.

6.4.2 Upper bounds

Let $\mathsf{ZTIME}(T(n))$ be the set of languages decidable by a zero-error probabilistic Turing machine that runs in expected time $O(T(n))$ for any input of length n.

6.4.6. Theorem. $\mathsf{CSPACE}(s(n)) \subseteq \mathsf{ZTIME}(2^{O(s(n))})$.

Proof. Consider an input x of length n, and let $O(s(n))$ be the available space on the work tape and s_a be the size of the auxiliary tape of the machine M. Since the total space available to the catalytic machine equals $s + s_a$, it has at most $O(2^{s+s_a})$ possible configurations. We take s_a to be at most $2^{O(s)}$.

When running M with input x and auxiliary start w, the machine can visit any configuration only once, since otherwise it would never halt. Similarly, a catalytic Turing machine can also not have any configuration in common between a computation starting with w or one with $w' \neq w$, for a certain input x; from that point on they would run the same computation, so the restored auxiliary part at halting would be incorrect for at least one of them.

Because of this uniqueness property, we can bound the expected runtime of a catalytic computation by simple counting. Note that the total number of different configurations that a Turing machine of memory $s + s_a$ can have is bounded by $O(2^{s_a+s+\log s_a+\log s})$, where we need the logarithmic terms to account for the location of the tape heads. Let $\textsc{time}\, M(x, w)$ denote the computation time of M on input x with the auxiliary tape initialized to w. Then it holds that

$$\sum_{w=0}^{2^{s_a}-1} \textsc{time}\, M(x, w) \leq O(2^{s_a+s+\log s_a+\log s})\,.$$

Dividing by 2^{s_a} gives

$$\mathop{\mathbb{E}}_{w \in_R \{0,1\}^{s_a}} [\textsc{time}\, M(x, w)] \leq 2^{O(s)},$$

where we use that $\log s_a = O(s)$. Now the inclusion in $\mathsf{ZTIME}(2^{O(s(n))})$ directly follows: a simulating zero-error probabilistic machine can just run the same computation as M, randomly generating bits of w as needed, and halt in expected time $2^{O(s)}$. □

In particular, for catalytic logspace, $\mathsf{CL} = \mathsf{CSPACE}(\log n) \subseteq \mathsf{ZPP}$.

A natural question to ask is: can a catalytic machine directly simulate deterministic Turing machines that use strictly more space, by having a translation for every instruction? From the previous theorem it follows that the answer is no. (Lack of this type of simulation of course does not rule out the possibility that the catalytic machine could decide languages that need more space, it only hints that such a construction can not use another Turing machine as a black box.)

6.4.7. Corollary. *No step-by-step simulation of deterministic space $\omega(s(n))$ is possible in catalytic space $s(n)$.*

Proof. There is some computation M on space $\omega(s(n))$ that uses time $t = 2^{\omega(s(n))}$ for all inputs of length n. Let x be an input of length n. Suppose that M has a step-by-step catalytic simulation M', which runs in space $s = s(n)$ with auxiliary space s_a.

By the definition of a step-by-step simulation, we have that

$$\forall a \in \{0,1\}^{s_w} \, \textsc{time}\, M'(x,w) \geq \textsc{time}\, M(x) \geq 2^{\omega(s)}.$$

From the proof of Theorem 6.4.6 we know that on expectation over w, M' must have $\textsc{time}\, M'(x,w) \leq O(2^s)$, a contradiction. □

6.4.8. Corollary. *If* $\mathsf{ZPP} = \mathsf{L}$ *then* $\mathsf{CSPACE}(s(n)) = \mathsf{DSPACE}(s(n))$.

Proof. The statement immediately follows from Theorem 6.4.6. Using padding, $\mathsf{ZPP} = \mathsf{L}$ implies $\mathsf{ZTIME}(2^{s(n)}) \subseteq \mathsf{DSPACE}(s(n))$, which gives $\mathsf{CSPACE}(s(n)) \subseteq \mathsf{ZTIME}(2^{s(n)}) \subseteq \mathsf{DSPACE}(s(n))$. □

6.4.9. Corollary. *The exponential-time hypothesis [IP99] implies that* $\mathsf{SAT} \notin \mathsf{CSPACE}(o(n))$.

Proof. The ETH says that $\mathsf{SAT} \notin \mathsf{BPTIME}(2^{o(n)})$. From this it directly follows that $\mathsf{SAT} \notin \mathsf{ZTIME}(2^{o(n)})$ and by Theorem 6.4.6 this implies $\mathsf{SAT} \notin \mathsf{CSPACE}(o(n))$. □

6.4.3 Oracle results for catalytic computation

We can show an oracle relative to which $\mathsf{CL} = \mathsf{CSPACE}(\log n) = \mathsf{PSPACE}$.

6.4.10. Theorem. *There exists an oracle A such that*

$$\mathsf{DSPACE}^A(2^{\Omega(s(n))}) = \mathsf{CSPACE}^A(s(n))$$

The intuition behind the proof is as follows. Any auxiliary string is either compressible, in which case we can replace it by a compressed version and use the now-available free space, or hard to compress, in which case we can make some non-trivial use of it — in this case as a 'password' for the oracle that can not be found by a small-space computation.

Some care has to be taken when interpreting oracle results for space-bounded computation. For example, there are oracles relative to which classic results like Savitch's theorem and the Immerman-Szelepcsènyi theorem do not hold.

Proof. Kolmogorov complexity will give us the notion of compressibility:

6.4.11. Definition. Fix some choice U for a universal Turing machine, and let x, y be two binary strings. The Kolmogorov complexity of a x relative to y, denoted $C(x|y)$ is the size of the smallest program p for machine U that outputs x on input y (i.e., $U(p,y) = x$). The Kolmogorov complexity of x, denoted $C(x)$, is $C(x|\varepsilon)$.

6.4.12. Fact. [Chain Rule [ZL70]] $C(x,y) \geq C(x) + C(y|x) - 4 \log C(x,y) - O(1)$.

We will construct an oracle A such that, relative to this oracle, a catalytic computation with work-tape space $s = s(n)$ can simulate a deterministic computation that uses space $2^{s/16}$. As a minor technical restriction, consider $s(n)$ such that $2^{s(n)/8} = \omega(n)$, i.e., $s(n)$ is at least $c \log(n)$ for $c > 8$.

Let w be a bit-string of length 2^s, the arbitrary initial contents of the auxiliary tape.

The oracle A will be given by four distinct *parts*, which we first describe informally. The first part checks if the (relative) Kolmogorov complexity of a given string is low. The second and third part can be respectively used to compress or decompress a given string. The fourth part, for which the definition is slightly more involved, gives access to a complete set for the large space computation when given a string with high complexity.

$$\begin{aligned}
A_1 &= \Big\{\langle 1, s, w, w'\rangle \Big| \, |w| = 2^{s/8} \text{ and } C(w|w') < \frac{3}{4}s\Big\} \\
A_2 &= \Big\{\langle 2, w, w', i, b\rangle \Big| b \text{ is the } i\text{-th bit of the smallest } p \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{such that } U(p, w') = w\Big\} \\
A_3 &= \Big\{\langle 3, w, p, i, b\rangle \Big| b \text{ is the } i\text{-th bit of } U(p, w)\Big\} \\
A' &= A_1 \cup A_2 \cup A_3
\end{aligned}$$

Now let $K^{O}_{f(n)}$ be a complete language for space $f(n)$ relative to oracle O. We define A_4 in stages, where the complete set is given relative to only the previous stages.

$$\begin{aligned}
A_4^{(n)} &= \Big\{\langle 4, w, x\rangle \;\Big|\; |x| = n \text{ and } C(a) \geq 2^{s(n)/8} \text{ and } x \in K^{A' \cup A_4^{<n}}_{2^{s(n)/16}}\Big\} \\
A_4 &= \bigcup_n A_4^{(n)}
\end{aligned}$$

Here $A_4^{<n} = \bigcup_{i=1}^{n-1} A_4^{(i)}$. Now the oracle A is the union of these parts, $A = A' \cup A_4$.

Let us give an algorithm to decide any given language $L \in \mathsf{DSPACE}^{A}(2^{s(n)/16})$. We divide the first $2^{s/4}$ bits of w into $2^{s/8}$ parts each of size $2^{s/8}$ and name the parts $w_1, \ldots, w_{2^{s/8}}$. Let $w_{<i}$ be the concatenation of w_1 up to w_{i-1}.

Starting with $i = 1$, ask part 1 of the oracle if $C(w_i|w_{<i}) < \frac{3}{4}s$. If that is not the case, increment i and repeat. If that *is* the case, then use the second part of the oracle to find the compressed version of w_i (given $w_{<i}$). Then store the compressed string version in our ordinary memory of size s, and erase the w_i part in the auxiliary tape. This frees up $2^{s/8}$ bits of memory, which we can use to decide if $x \in L$. When we are done with that, we can use the third part of the oracle to decompress w_i back into the auxiliary tape.

If none of the w_i for $i \in \{1, \ldots, 2^{s/8}\}$ are compressible given $w_{<i}$, we can show a lower bound for the Kolmogorov complexity of w using the chain rule:

$$\begin{aligned} C(w_1, w_2, \ldots, w_{2^{s/8}}) &\geq \sum_{i=1}^{2^{s/8}} (C(w_i | w_{<i}) - 4 \log C(w) - O(1)) \\ &\geq 2^{s/8} \Big(\frac{3}{4} s - \frac{4}{8} s - O(1) \Big) \\ &\geq 2^{s/8} \end{aligned}$$

(for s sufficiently large). Now we can use w as a high complexity 'password' for the fourth part of the oracle.

No machine in space $o(2^{s/8})$ can make a query of complexity as large as w. To see this, consider the configuration of the machine (including the input tape) before it starts writing the first character of any query q to the oracle tape. This configuration can be stored using $O(2^{s(n)/16}) + n = o(2^{s(n)/8})$ bits, but it contains all the information needed to produce q — a contradiction if q has Kolmogorov complexity at least $2^{s/8}$.

This implies that machines with space $2^{s(n)/16}$, on an input of length n, cannot distinguish $A' \cup A_4^{<n}$ from A, because they cannot query any string in $A_4^{(i)}$, for $i \geq n$. For any n it then holds that $K_{2^{s(n)/16}}^{A' \cup A_4^{<n}} = K_{2^{s(n)/16}}^{A}$, for the accessible strings of length n, and hence, having access to the string a and the oracle A, our catalytic machine can decide $K_{2^{s(n)/16}}^{A}$ (and therefore whether $x \in L$) by using the part 4 of the oracle. □

6.4.13. THEOREM. *There is an oracle B such that $\mathsf{NL}^B \not\subseteq \mathsf{CSPACE}^B(\log n)$.*

Proof. A Baker–Gill–Solovay [BGS75] construction works: from the proof of Theorem 6.4.6 we know that a Turing machine M deciding a language in $\mathsf{CSPACE}(\log n)$ has to run in average polynomial time, averaged over all possible auxiliary starting contents w. Therefore for any input x there is always a w for which M makes only polynomially many queries, and we apply the construction for that starting state — we diagonalize against the machine M at a string in the oracle that is not queried by $M(x, w)$. Because the outcome of the catalytic computation should be correct for all possible starting values, the existence of a value w such that the machine fails implies that the machine does not correctly decide the language. □

Chapter 7

Catalytic computation: Non-determinism and hierarchy

Catalytic computation, which we defined in the previous chapter (first published as [BCK$^+$14]), is a space-bounded computation where in addition to our working memory we have an exponentially larger auxiliary memory which is full; the auxiliary memory may be used throughout the computation, but it must be restored to its initial content by the end of the computation.

Motivated by the surprising power of this model, we set out to study the non-deterministic version of catalytic computation. We establish that non-deterministic catalytic logspace is contained in ZPP, which is the same bound known for its deterministic counterpart, and we prove that non-deterministic catalytic space is closed under complement (under a standard derandomization assumption). Furthermore, we establish hierarchy theorems for non-deterministic and deterministic catalytic computation.

The results in this chapter are based on:

- [BKLS16] Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. Catalytic space: non-determinism and hierarchy. In *33rd Symposium on Theoretical Aspects of Computer Science* (STACS 2016).

7.1 Introduction

In the previous chapter, joint work which was published as Buhrman et al. [BCK$^+$14], we defined the notion of *catalytic computation*, a space-bounded model of computation in which the usual Turing machine has, in addition to its work tape, access to a large auxiliary memory which is full. The auxiliary memory can be used during the computation, but its starting contents must be restored by the end of the computation. The space usage that is counted is the amount of work space s used; the auxiliary memory is for free. In a reasonable setting, the auxiliary memory is of size at most 2^s. One can think of the auxiliary memory as a hard disk full of data. The catch with the auxiliary memory is that it may contain arbitrary content, possibly incompressible, which has to be preserved in some way during the computation. It is not obvious whether such auxiliary memory can be useful at all. In the previous chapter we showed that, surprisingly, there is a non-trivial way of using the full memory; that it is possible to compute in work space $O(\log n)$ (*catalytic log-space,* CL) functions not known to be computable in the usual logarithmic space (*log-space,* L without the auxiliary memory. Indeed, all of TC^1, which includes NL and LOGCFL, is contained in CL.

This motivated us to explore further: What other problems can be solved in catalytic log-space? Buhrman et al.[1] show $\mathsf{CL} \subseteq \mathsf{ZPP}$, so CL is unlikely to contain the whole of PSPACE (even though this is the case relative to some oracle). The fact that $\mathsf{NL} \subseteq \mathsf{CL}$ suggests an obvious question: what about non-deterministic catalytic log-space? Could it be that non-deterministic computation equipped with auxiliary tape has the same power as deterministic catalytic computation? Non-deterministic catalytic computation could possibly allow us to identify further problems that can benefit from having full memory. The previous chapter also raises a host of further question about the catalytic model such as: Is there a space hierarchy? Does some kind of Savitch's theorem hold for catalytic log-space? Is non-deterministic catalytic space closed under complement? *etc.* The work we present in this chapter aims to shed light on some of these questions.

In this chapter we show that non-deterministic catalytic space is closed under complement under a widely accepted derandomization assumption. We also establish hierarchy theorems for catalytic computation in the deterministic and non-deterministic settings. For our non-deterministic catalytic log-space we can also establish the same ZPP upper bound that was known for CL. Hence there seems to be a closeness between determinism and non-determinism for catalytic computation. Despite that we are unable to establish an equivalent of Savitch's theorem. This remains an intriguing open problem.

We prove the closure under complement using the inductive counting technique of Immerman and Szelepcsényi [Imm88, Sze88]. However, we had to overcome several difficulties. One challenge is that we might be faced with an exponential-size graph of reachable configurations. We show how to use a pseudorandom generator to avoid such a situation. Another issue is that for inductive counting we need to be able to remember and reason about different configurations. However, the full description of a configuration is exponentially bigger than our work space, so we cannot possibly store it in full. This is one of the hurdles that prevents us from carrying out Savitch's algorithm for catalytic computation. For the inductive counting we resolve this issue by using fingerprints for various configurations.

Our hierarchy theorems are proven in the setting of computation with advice. The catalytic model is a semantic restriction. It is an easy exercise to show that it is algorithmically undecidable whether a machine will restore the full memory on every input to its original content. For semantic models of computation, like bounded-error randomized computation, the only hierarchy theorems that we know of are in the setting with advice. The reason is that essentially all known hierarchy theorems are proven by diagonalization, which requires the ability to enumerate exactly all machines of a given type. We do not know any such enumeration for catalytic machines so we have to settle for the weaker result. The advice is used only to tell the diagonalizing machine whether it is safe to diagonalize against a particular machine. The hierarchy theorems follow from the work of Kinne and van Melkebeek, and van Melkebeek and Pervyshev [KvM10, vMP06]. For some space bounds we provide more accurate separations that were not explicitly calculated before.

---

[1] Presented as Theorem 6.4.6 in this thesis.

The layout of the chapter is as follows. Section 7.2 contains some preliminaries. In Section 7.3 we define non-deterministic catalytic computation, and prove that the corresponding log-space class CNL is contained in ZPP. Section 7.4 is devoted to proving that CNL is closed under complement, and in Section 7.5 we show hierarchy theorems for catalytic computation.

7.2 Preliminaries

We gave a short introduction to complexity theory in Section 2.4, but for convenience we will quickly mention the complexity classes that are relevant for the current chapter.

The complexity class L denotes the problems solvable in logspace, while PSPACE is the class of those problems that can be solved using a polynomial amount of space. The class NL contains the problems that can be solved *non-deterministically* in logspace, and LOGCFL is the class of problems that are logspace many-one reducible to context-free languages.

The problems in ZPP (zero-error probabilistic polynomial time) are the ones computable by a probabilistic Turing machine, that halts in expected polynomial time, while always outputting the correct answer for any input.

One circuit class we mention is TC^1, which is the class of boolean functions computable by circuits of depth $O(\log n)$ by AND gates, OR gates and MAJ gates, all with unbounded fan-in — a MAJ gate outputs 1 if and only if most of its input bits are 1. We use $\mathsf{SIZE}(s)$ to denote the class of problems that can be solved by circuits of size s.

We introduced the notion of a catalytic computation in the previous chapter. Since we will define the non-deterministic variant shortly, we quickly summarize the definition here for easy comparison.

7.2.1. Definition. Let M be a deterministic Turing machine with four tapes: one input and one output tape, one work tape, and one *auxiliary tape* (or *aux-tape*).

M is said to be a *catalytic Turing machine* using workspace $s(n)$ and auxiliary space $s_a(n)$ if for all inputs $x \in \{0,1\}^n$ and auxiliary tape contents $w \in \{0,1\}^{s_a(n)}$, the following three properties hold [BCK+14].

1. **Space bound.** The machine $M(x,w)$ uses space $s(n)$ on its work tape and space $s_a(n)$ on its auxiliary tape.
2. **Catalytic condition.** $M(x,w)$ halts with w on its auxiliary tape.
3. **Consistency.** The outcome of the computation $M(x,w)$ is consistent among all initial aux-tape contents w.[2]

We used this to obtain an analogue of the usual space-bounded complexity classes:

7.2.2. Definition. $\mathsf{CSPACE}(s(n), s_a(n))$ is the class of decision problems solvable by a catalytic Turing machine using workspace $O(s(n))$ and auxiliary space $O(s_a(n))$. The notational shorthand $\mathsf{CSPACE}(s(n))$ is defined as the complexity class $\mathsf{CSPACE}(s(n), 2^{O(s(n))})$. The class CL is $\mathsf{CSPACE}(\log n)$.

In the previous chapter (published as [BCK+14]), it was shown that, surprisingly, CL can make a non-trivial use of the auxiliary tape. Indeed, we have shown that $\mathsf{TC}^1 \subseteq \mathsf{CL}$, but it is generally believed that $\mathsf{TC}^1 \not\subseteq \mathsf{L}$.

In this chapter, we will first prove an analogue of the Immerman–Szelepcsényi theorem. The definition of the non-deterministic version of CL, denoted CNL, will be left for Section 7.3. Then $\mathsf{CNL} = \mathsf{coCNL}$ will hold under the same assumption as the following standard derandomization result, whose proof is now standard.[3]

7.2.3. Lemma ([IW97, KvM02]). *If there exists a constant* $\varepsilon > 0$ *such that* $\mathsf{DSPACE}(n) \not\subseteq \mathsf{SIZE}(2^{\varepsilon n})$ *then for all constants c there exists a constant* c' *and a function* $G : \{0,1\}^{c' \log n} \to \{0,1\}^n$ *such that for any circuit* $\mathcal{C}$ *of size* n^c

$$\left| \Pr_{r \in \{0,1\}^n}[\mathcal{C}(r) = 1] - \Pr_{s \in \{0,1\}^{c' \log n}}[\mathcal{C}(G(s)) = 1] \right| < \frac{1}{n}$$

and G is computable in space logarithmic in n.

We will also need a hash family with nice properties.

---

[2] What this means depends on what we are trying to do. For instance, when solving a decision problem, $M(x,w)$ should either accept for all choices of w — in which case we say M accepts x — or it rejects for all possible w — M rejects x.

[3] For instance, the pseudo-random generator of [IW97] has the right properties. Also see Appendix C of [KvM02] and Theorem 19 of [ABK+06].

7.2.4. LEMMA. *For every n, there exists a family of hash functions $\{h_k\}_{k=1}^{n^3}$, with each h_k a function $\{0,1\}^n \to \{0,1\}^{4\log n}$, such that for every k the following properties hold. Firstly, h_k is computable in space $O(\log n)$, and secondly, for every set $S \subset \{0,1\}^n$ with $|S| \leq n$ there is a hash function in the family that is injective on S.*

The proof of this lemma is a simple exercise; we do include it for completeness at the end of the current section.

In addition to studying non-determinism, we will prove a space-hierarchy theorem for catalytic computations. This hierarchy theorem holds for catalytic Turing machines with an advice string.

We define advice added to a catalytic computation in the same way as in the recent line of research that proves hierarchies for certain classes of semantic models, see for example [vMP06, KvM10]. In our case that means that a computation needs to satisfy the catalytic condition and consistency properties on the *correct* advice, and is allowed to (for example) fail to restore the contents of the aux-tape for other values of the advice. This notion of advice is a variation on the one defined by Karp and Lipton [KL82], who required that the machine model was robust under all possible values of the advice string. Proving the same hierarchy theorem using the Karp-Lipton definition would be harder, and would indeed imply a hierarchy theorem that also holds without any advice [KvM10].

The proof of the hierarchy for catalytic computation uses Savitch's theorem [Sav70], a classic result in complexity theory, which shows that for any space bound $s(n)$ at least $\log n$ it holds that $\mathsf{NSPACE}(s(n)) \subseteq \mathsf{DSPACE}(s(n)^2)$. That is, any problem which can be solved by a non-deterministic space-bounded Turing machine, can also be solved by a deterministic Turing machine using a squared amount of space.

Remarks on notation. For two binary strings x, y of equal length, we use $x \oplus y$ for the bitwise XOR of x and y. The function log always stands for the logarithm of base 2. For simplicity, all Turing machines are assumed to use a binary alphabet — all definitions and proofs would easily generalize to larger alphabet sizes, at the cost of introducing notational clutter.

7.2.1 Existence of hash family

7.2.5. THEOREM (CHINESE REMAINDER THEOREM). *Let $p_1, \ldots, p_m$ be a list of relatively prime integers. Any positive integer x is uniquely specified by the list of remainders $a_1 = x \mod p_1, a_2 = x \mod p_2, \ldots, a_m = x \mod p_m$, provided that $x < \prod_{i=1}^m p_i$.*

Proof of Lemma 7.2.4. For a natural number k, let p_k be the k-th prime number. For every $k = 1, \ldots, n^3$ define the hash function $h_k(x) = x \mod p_k$. We will show that for any set $S \subset \{0,1\}^n$ of size n, there exists a number $k^* \in \{1,\ldots,n^3\}$ such that the function h_{k^*} is injective on S. Here we interpret binary strings as natural numbers in the usual way, and hence we can upper bound any element of S by 2^n.

For all $x, y \in S$, where $x \neq y$, define $B_{x,y} = \{p_k \mid x \bmod p_k = y \bmod p_k, 1 \leq k \leq n^3\}$ to be the set of primes for which x and y hash to the same value. Then $B = \bigcup_{x,y\in S, x\neq y} B_{x,y}$ is the set of all primes which give a hash collision on the set S.

For any pair x, y it now holds that $|B_{x,y}| \leq n$. Indeed, assume for a contradiction that the set contains a subset of $n+1$ primes for which x and y have the same remainders. Noting that the product of these $n+1$ primes is at least 2^n, larger than both x and y, and that prime numbers are relatively prime, we find an immediate contradiction with the Chinese Remainder Theorem.

We can bound the number of primes that give a collision by

$$|B| \leq \sum_{x,y\in S, x\neq y} |B_{x,y}| \leq \binom{n}{2} n\,,$$

which is strictly less than n^3 for $n > 1$. Therefore there exists a prime p_k with $1 \leq k \leq n^3$ such that $p_k \notin B$, and therefore $x \bmod p_k$ is unique for all $x \in S$.

Left is to show this algorithm can be executed in logarithmic space. First note that using the prime number theorem we can (imprecisely) bound $p_k \leq n^4$, for $1 \leq k \leq n^3$. Since every number $p \leq n^4$ we try as modulus can be stored using $4 \log n$ bits, checking primality is also readily seen to be in space $O(\log n)$, just by checking all possible factors. To hash a value $x \in \{0,1\}^n$ we can, for example, sum $2^i \bmod p$ for all i such that $x_i = 1$. The value $2^i \bmod p$ can easily be computed in space $O(\log n)$ by repeated multiplication by 2, i.e., a bit shift, followed by subtraction of p whenever the intermediate value becomes too large. $\square$

7.3 Non-deterministic catalytic computation

The model for catalytic computation is defined in terms of deterministic Turing machines. This gives rise to the question: What would the power of a non-deterministic version of CL be? In this section we extend the definitions of catalytic-space computation to the non-deterministic case, and prove basic results about this model.

7.3.1. Definition. Let M be a non-deterministic Turing machine, with four tapes: one input and one output tape, one work-tape, and one *auxiliary tape*.

Let $x \in \{0,1\}^n$ be an input, and $w \in \{0,1\}^{s_a(n)}$ be the initial contents of the auxiliary tape. We say that $M(x,w)$ accepts x if there exists a sequence of nondeterministic choices that makes the machine accept. If for all possible sequences of nondeterministic choices $M(x,w)$ does not accept, the machine rejects x.

Then M is said to be a *catalytic non-deterministic Turing machine* using workspace $s(n)$ and auxiliary space $s_a(n)$ if for all inputs, the following three properties hold.

1. **Space bound.** The machine $M(x, w)$ uses space $s(n)$ on its work tape and space $s_a(n)$ on its auxiliary tape.
2. **Catalytic condition.** $M(x, w)$ halts with w on its auxiliary tape, irrespective of its nondeterministic choices.
3. **Consistency.** The outcome of the computation $M(x, w)$ is consistent among all initial aux-tape contents w. This means that for any given input x, $M(x, w)$ should always accept, or always reject, regardless of w; *however*: the specific nondeterministic choices that make $M(x, w)$ go one way or the other may depend on w.

7.3.2. Definition. $\mathsf{CNSPACE}(s(n), s_a(n))$ is the class of decision problems solvable by a catalytic non-deterministic Turing machine using workspace $s(n)$ and auxiliary space $s_a(n)$, and $\mathsf{CNSPACE}(s(n))$ is defined as $\mathsf{CNSPACE}(s(n), 2^{s(n)})$. The class CNL is $\mathsf{CNSPACE}(O(\log n))$.

We now have an analogue of non-deterministic space-bounded complexity. For convenience, as an appendix to the chapter we present an equivalent definition of CNL with all the conditions unfolded.

A note on alternative definitions. There are multiple possible ways to add non-determinism to a catalytic Turing machine. For instance, we require the machine to restore the contents of the auxiliary tape for any given sequence of non-deterministic bits; but at a first glance, it seems we could make this requirement only for those non-deterministic guesses which result in accepting states. However, defining the model in this way is less natural for several reasons. For one, we can not run two machines sequentially and accept if one of them accepts: if one of the two machines would reject, the whole computation needs to reject, because the auxiliary tape may have been irreversibly changed; so the class would not be closed under union. This would also prevent amplification of success probability in a probabilistic class defined using such machines. Philosophically speaking, having a catalytic machine which ‘sometimes’ destroys all data it is guaranteed to preserve, seems to go against the spirit of the model.

Another possible variation would be to require that the accepting sequence of non-deterministic choices is independent of the initial contents of the auxiliary tape, which would give a weaker model. Indeed, this would not look very strange in a certificate definition, effectively requiring that there exists a read-once certificate, independent of the initial contents of the aux-tape, which can be verified by a deterministic log-space catalytic Turing machine. Even so, when describing the model with non-deterministic Turing machines it seems unnatural to have this restriction. Furthermore, the model is weaker, so if we expect to make some use of non-determinism, it should be easier if we define it in the current way. Hence we have also ruled out this alternative definition.

7.3.1 Simulation by probabilistic computation

In Theorem 6.4.6, we proved that $\mathsf{CL} \subseteq \mathsf{ZPP}$; we now generalize this to our new non-deterministic model by showing that $\mathsf{CNL} \subseteq \mathsf{ZPP}$.

7.3.3. Definition. Define the directed acyclic graph $\mathcal{G}_{M,x,w}$ to be the configuration graph of a catalytic non-deterministic Turing machine M on input x and auxiliary tape starting contents w. That is, $\mathcal{G}_{M,x,w}$ has a node for every configuration which is reachable by non-deterministic choices when executing $M(x,w)$.

We will use $|\mathcal{G}_{M,x,w}|$ to denote the number of nodes of the configuration graph.

7.3.4. Lemma. *Let M be a non-deterministic catalytic machine using space $c \log n$ and let $c' = 2c + 2$. Then for all x*

$$\mathop{\mathbb{E}}_{w \in_R \{0,1\}^{n^c}} \left[|\mathcal{G}_{M,x,w}| \right] \leq O(n^{c'}) .$$

Proof. Notice that, for any given $x \in \{0,1\}^n$, and for different auxiliary tape contents w, w', the set of configurations in $\mathcal{G}_{M,x,w}$ and in $\mathcal{G}_{M,x,w'}$ have to be disjoint. For the sake of contradiction, consider a configuration q that is reachable both by $M(x,w)$ and by $M(x,w')$. Then any halting configuration reachable by q will have the wrong contents on its auxiliary tape for either the computation that started with w or with w'.

The number of bits needed to describe a configuration of M, excluding the contents of the input tape, is bounded by

$$c \log n + n^c + \log n^c + \log n + \log (c \log n) + O(1) \leq (2c+2) \log n + n^c + O(1),$$

where we do include the encoding of the location of the tape heads and the internal state of the Turing Machine. Therefore the total number of reachable configurations, counted over all possible starting auxiliary tape contents, is at most

$$\sum_{w \in \{0,1\}^{n^c}} |\mathcal{G}_{M,x,w}| \leq 2^{c' \log n + n^c + O(1)} = O(n^{c'}) 2^{n^c}$$

And thus:

$$\frac{1}{2^{n^c}} \sum_{w \in \{0,1\}^{n^c}} |\mathcal{G}_{M,x,w}| = \mathop{\mathbb{E}}_{w \in_R \{0,1\}^{n^c}} \left[|G_{M,x,w}| \right] \leq O(n^{c'}) .$$

□

Now suppose we have CNL machine M, and let $x \in \{0,1\}^n$ be the input string. Consider an algorithm which flips a random string w and searches $\mathcal{G}_{M,x,w}$ for a path from the initial configuration to an accepting configuration. This takes time polynomial in $|G_{M,x,w}|$. By Lemma 7.3.4 this graph is polynomial-sized in expectation, and therefore this procedure finishes in expected polynomial time. Thus we obtain:

7.3.5. Corollary. $\mathsf{CNL} \subseteq \mathsf{ZPP}$.

7.4 An analogue of the Immerman–Szelepcsényi theorem

This section is devoted to proving that CNL is closed under complement. Our proof strategy is based on the inductive-counting argument to prove the Immerman–Szelepcsényi theorem. In order for the proof to work for catalytic computation, we will need a couple of new ideas.

Suppose we are given a CNL machine M, and wish to construct a CNL-machine M' to compute the complement M, via an inductive-counting argument on the configuration graph of M.

First of all, notice that whenever M' wishes to simulate a run of M, it must necessarily use its own aux-tape to simulate the aux-tape of M, because it is the only read-write tape that is big enough.

Now, for some w (initial contents of the aux-tape), M may visit exponentially many configurations. Then the inductive counting would be impossible to do with only logarithmic space. So the first idea is to use the pseudo-random generators of Lemma 7.2.3 to avoid such *bad* w. Lemma 7.4.1 explains why this works.

Notice also that we must be careful that M', when simulating a run of M, can always restore the initial contents of its aux-tape. We can make sure this happens correctly by using the catalytic condition applied to M: whenever we need to restore the initial contents of the aux-tape, it will be enough to run the simulation of M to an arbitrary halting configuration.

Finally, recall that the inductive-counting argument involves storing and comparing configurations of M; but the configurations of M include the aux-tape, and are too big for the simulating machine M' to store on its work tape. So the second idea is to use the family of hash functions of Lemma 7.2.4, and do inductive-counting by storing and comparing *the hashes* of configurations instead.

Putting the whole thing together, however, is rather delicate, because our pseudo-random generator will still give us bad w's for some seeds, and there is no easy CNL way of showing that a given hash function is collision free on the set of reachable configurations of a CNL machine (that is a coCNL predicate); but our algorithm has to manage anyway.

Let us start by showing how to avoid bad w's.

7.4.1. Lemma. *Assume the derandomization condition of Lemma 7.2.3, and let G be as given therein. Let M be a non-deterministic catalytic Turing machine using workspace $c \log n$. Then, for every input x and aux-tape contents w, at least half of the seeds $s \in \{0,1\}^{O(\log n)}$ will cause the non-deterministic computation $M(x, G(s) \oplus w)$ to reach at most n^{2c+3} many different configurations.*

Proof. Let M be a CNL machine using workspace $c \log n$ and auxiliary space n^c. Let $x \in \{0,1\}^n$, $w \in \{0,1\}^{n^c}$ be given.

Let $C_{x,w}$ be a boolean circuit which, on input $r \in \{0,1\}^{n^c}$, does a breadth-first traversal of $\mathcal{G}_{M,x,r\oplus w}$[4], starting on the initial configuration, until either:

i. More than n^{2c+3} nodes have been found, in which case it outputs 0; or

ii. The graph has been fully traversed, in which case it outputs 1.

The size of $C_{x,w}$ can be bounded by a polynomial, say n^d. The circuit $C_{x,w}$ outputs 1 on input r if and only if $|\mathcal{G}_{M,x,r\oplus w}| \leq n^{2c+3}$. Therefore, for large enough n, for all $x \in \{0,1\}^n$ and all $w \in \{0,1\}^{n^c}$,

$$\begin{aligned}
\Pr_{r \in_R \{0,1\}^{n^c}}[\mathcal{C}_{x,w}(r) = 0] &= \Pr_{r \in_R \{0,1\}^{n^c}}\Big[|\mathcal{G}_{M,x,r\oplus w}| \geq n^{2c+3}\Big] \\
&= \Pr_{r \in_R \{0,1\}^{n^c}}\Big[|\mathcal{G}_{M,x,r}| \geq n^{2c+3}\Big] \\
&\leq \frac{1}{n^{2c+3}} \mathop{\mathbb{E}}_{r \in_R \{0,1\}^{n^c}}\Big[|\mathcal{G}_{M,x,r}|\Big] \\
&\leq O\Big(\frac{1}{n}\Big).
\end{aligned}$$

Here we have used the fact that, for a fixed w, r and $r \oplus w$ are equidistributed. The last inequality follows from Markov's inequality and Lemma 7.3.4.

Now Lemma 7.2.3 provides us with a log-space computable function $G : \{0,1\}^{O(\log n)} \to \{0,1\}^{n^c}$ such that, for all $x \in \{0,1\}^n$ and $w \in \{0,1\}^{n^c}$,

$$\left| \Pr_{r \in \{0,1\}^{n^c}}[\mathcal{C}_{x,w}(r) = 0] - \Pr_{s \in \{0,1\}^{O(\log n)}}[\mathcal{C}_{x,w}(G(s)) = 0] \right| \leq \frac{1}{n}.$$

In particular, for all sufficiently large n we get the rough bound:

$$\Pr_{s \in \{0,1\}^{O(\log n)}}[\mathcal{C}_w(x, G(s)) = 0] \leq \frac{1}{n} + O\Big(\frac{1}{n}\Big) < \frac{1}{2}.$$

Therefore, for any x and w, at least half of the seeds s will ensure that the configuration graph $\mathcal{G}_{M,x,G(s)\oplus w}$ has at most n^{2c+3} nodes. □

---

[4] Recall that $\mathcal{G}_{M,x,r\oplus w}$ is the configuration graph of M, for input x and aux-tape contents given by the bit-wise XOR of r and w.

Our goal is now to use an inductive counting argument on $\mathcal{G}_{M,x,G(s)\oplus w}$. Like we mentioned earlier, inductive counting requires us to write down configurations in the work tape, but the tape is not big enough. To circumvent this, we will instead write down the *hash values* of the configurations, via the hash family of Lemma 7.2.4. The proof below puts it all together.

7.4.2. THEOREM (IMMERMAN–SZELEPCSÉNYI FOR CATALYTIC COMPUTATION). *If there exists a constant* $\varepsilon > 0$ *such that* $\mathsf{DSPACE}(n) \not\subseteq \mathsf{SIZE}(2^{\varepsilon n})$ *then* $\mathsf{CNL} = \mathsf{coCNL}$.

Proof. Let M be a nondeterministic Turing machine that uses $d \log n$ work space, and has an auxiliary tape of size n^d. We wish to construct a nondeterministic catalytic Turing machine M', using workspace $O(\log n)$, such that for any n and any input $x \in \{0,1\}^n$ our computation accepts x if M rejects x, and vice-versa.

Without loss of generality, assume that for any given $w \in \{0,1\}^{n^d}$, $M(x,w)$ has a unique accepting configuration acc_w. Let $start_w$ be the initial configuration of $M(x,w)$ and let $e = 2d + 3$.

By the consistency property, either there exists a path from $start_w$ to acc_w for all w, or it is impossible to reach acc_w from $start_w$, for any w. We prove Theorem 7.4.2 by describing a way of certifying that there exists no path between $start_w$ and acc_w in $\mathcal{G}_{M,x,w}$.

Fix some input x, and let w' denote the initial contents of the aux-tape of M'. By Lemma 7.4.1, we know that for at least half of the possible seeds $s \in \{0,1\}^{O(\log n)}$, we have

$$|\mathcal{G}_{M,x,G(s)\oplus w'}| \leq n^e \,. \tag{7.1}$$

If (7.1) holds, we say s is a *good seed.*

Lemma 7.2.4 gives us a family of hash functions $\{h_k\}_{k=1}^{n^{3e}}$, with the property that, for every good seed s, there is at least one hash function in the family which is one-to-one on the nodes of $\mathcal{G}_{M,x,w}$.

Below, as Algorithm 7.4, we give the pseudo-code for M''s algorithm. Let us now do a guided reading of this code. We begin by breaking the code into three sections, for the lines 2–6, 7–26, and 27–32.

Algorithm 2 Pseudo-code for M'.

Here G is the log-space PRG of Lemma 7.2.3, S is the number of seeds, $m = n^e$ stands for the maximum number of configurations allowed in the configuration graph, and H is the size of the hash family given by Lemma 7.2.4. The aux-tape is represented by a variable w, whose initial value is w'. The lines that use non-determinism are marked with a $(*)$.

```
 1: procedure coCNL-Simulation(Input $x$, Aux-Tape $w \leftarrow w'$)
 2:     $N \leftarrow 0$
 3:     for $s = 0 \dots S$ do
 4:         $w \leftarrow G(s) \oplus w$
 5:         $g \leftarrow 0$
 6:         $\ell \leftarrow 0$
 7:         for $k = 1 \dots H$ do
 8:             $c \leftarrow 1$
 9:             for $i = 1 \dots m$ do
10:                 $c' \leftarrow 0$
11:                 for $v = 0 \dots m$ do
12:                     if canReach$(v, i, h_k)$ then                      ▷ $(*)$
13:                         $c' \leftarrow c' + 1$
14:                     else if cannotReach$(v, i, c, h_k)$ then           ▷ $(*)$
15:                         Do nothing
16:                     else
17:                         Jump to line 30
18:                     end if
19:                 end for
20:                 $c \leftarrow c'$
21:             end for
22:             if $c > g$ then
23:                 $g \leftarrow c$
24:                 $\ell \leftarrow k$
25:             end if
26:         end for
27:         if cannotReach$(h_\ell(acc_w), m + 1, g, h_\ell)$ then        ▷ $(*)$
28:             $N \leftarrow N + 1$
29:         end if
30:         $w \leftarrow G(s) \oplus w$
31:     end for
32:     Accept if $N > S/2$, and Reject otherwise
33: end procedure
```

In lines 2–6, we initialize a variable N to 0 (line 2), cycle through every seed s (line 3), XOR the contents of the aux-tape with $G(s)$ (line 4), and initialize two variables g and ℓ to 0 (lines 5 and 6).

Then, in lines 7–26, we have an inner loop that cycles through every hash function (line 7). Below we will prove:

Property I If the seed s is good, then **(I.a)** some sequence of non-deterministic bits will cause the inner loop to exit normally at line 27, with the promise that $g = |\mathcal{G}_{M,x,w}|$, and that h_ℓ is one-to-one on $\mathcal{G}_{M,x,w}$; and **(I.b)** any sequence of non-deterministic bits that fails this promise will exit the inner loop by jumping directly to line 30.

At line 27, we use the value of g and ℓ we have obtained to try and certify that acc_w is not reachable. If we succeed to do so, we increment N (line 28). Below we will also prove:

Property II If the seed s is good, $g = |\mathcal{G}_{M,x,w}|$, and h_ℓ is one-to-one on $\mathcal{G}_{M,x,w}$, then some sequence of non-deterministic bits will cause us to successfully certify that acc_w is not reachable if and only if $M(x, w)$ rejects.[5]

Before we move on to the next seed, we first restore the initial contents of the aux-tape, by once again XORing them with $G(s)$ (line 30).

Finally, the procedure accepts if and only if $N > S/2$ in line 32. Let us prove that, assuming Properties I and II, the procedure accepts if and only if $M(x, w)$ rejects. Lemma 7.4.1 ensures that more than half the seeds are good, and hence:

1. If $M(x, w)$ rejects: Property I ensures that, for each good seed s, some non-deterministic guess will cause us to reach line 27 with $g = |\mathcal{G}_{M,x,w}|$ and h_ℓ one-to-one on $\mathcal{G}_{M,x,w}$; then Property II ensures that some further guess will result in N being incremented; hence some overall non-deterministic guess will give $N > S/2$, and the procedure will accept in line 32.

2. If $M(x, w)$ accepts: Property I ensures that, for each good seed s, if we reach line 27, then $g = |\mathcal{G}_{M,x,w}|$ and h_ℓ one-to-one on $\mathcal{G}_{M,x,w}$, and thus, by Property II, N will not be incremented in line 28. If some non-deterministic guess fails to get us to line 27, then Property I tells us that the execution jumped directly to line 30, so N was again not incremented. Because no good seed will ever cause N to be incremented, $N < S/2$ and the procedure rejects in line 32.

So all we need to do is prove properties I and II. We first need to specify the canReach and cannotReach subroutines. Their correctness is easy to see from the description and pseudo-code.

The canReach(v, i, h_k) subroutine (see Algorithm 3) checks whether there is a node w in $\mathcal{G}_{M,xw}$, reachable within i steps, with $h_k(w) = v$.

---

[5] But if s, g or h_ℓ are not as assumed, we might get a false-positive, claiming that acc_w is not reachable when in fact it is.

Behavior of the canReach subroutine. If such a w exists, then some non-deterministic guess will cause the procedure to return TRUE, and, otherwise, every non-deterministic guess will return FALSE.

canReach non-deterministically works as follows: we guess a length $L \leq i$, and simulate M for L steps. After this, we hash the configuration M is currently in, and compare it to v. We will then return TRUE if and only if the two hashes are the same, but before we return, we finish the simulation of M until we reach a halting state, in order to restore the contents of the aux-tape.

Algorithm 3 The canReach subroutine.

The subroutine to check that a node hashing to v is reachable in at most i steps, given some hash function h_k.

```
procedure canReach($v$, $i$, $h_k$)
    $z \leftarrow 0$                    ▷ Workspace and internal state of simulated machine
    Non-deterministically guess $L \leq i$                                  ▷ (*)
    Simulate $M(x, w)$ using $z$ as workspace for $L$ steps                  ▷ (*)
    if $h_k(z, w') = v$ then
        $r \leftarrow$ TRUE
    else
        $r \leftarrow$ FALSE
    end if
    Continue simulation of $M(x, w)$ using $z$ and reach any halting state
    return $r$
end procedure
```

The cannotReach(v, i, c, h_k) subroutine (see Algorithm 4) checks that there is no node in $\mathcal{G}_{M,x,w}$ hashing to v and reachable within i steps, as long as c and h_k fulfill the promise that there are exactly c nodes in $\mathcal{G}_{M,x,w}$ that are reachable within $i - 1$ steps, and that h_k is one-to-one on $\mathcal{G}_{M,x,w}$.

Behavior of the cannotReach subroutine. If the hash v is unreachable within i steps and the given c, h_k obey the promise, then some non-deterministic guess will cause the procedure to return TRUE. If v *is* reachable and c, h_k obey the promise, every guess will return FALSE. *Furthermore*, if the hash v is unreachable within i steps, and c is smaller than the number of nodes in $\mathcal{G}_{M,x,w}$ that are reachable within $i - 1$ steps, then there is a non-deterministic guess that causes the procedure to return TRUE, even if h_k is not one-to-one.

Algorithm 4 The cannotReach subroutine.

The subroutine checking that a node hashing to v is *not* reachable within i steps, for hash function h_k, when given c, the number of nodes reachable in $i-1$ steps.

```
procedure cannotReach($v$, $i$, $c$, $h_k$)
    $h' \leftarrow -1$                                        ▷ Hash of previously seen node
    for $j = 1 \dots c$ do
        $z \leftarrow 0$                     ▷ Workspace and internal state of simulated machine
        Non-deterministically guess $L \leq i - 1$                                  ▷ (∗)
        Simulate $M(x, w)$ using $z$ as workspace for $L$ steps                     ▷ (∗)
        if $h_k(z, w) \leq h'$ then                      ▷ Visited the nodes in wrong order
            Simulate $M(x, w)$ using $z$ and reach any halting state
            return FALSE
        end if
        $h' \leftarrow h_k(z, w)$
        while there are unvisited neighbours do
            Step $M(x, w)$ with workspace $z$ into a neighbour configuration
            if $h_k(z, w') = v$ then                          ▷ $v$ is reachable in $i$ steps
                Simulate $M(x, w)$ using $z$ and reach any halting state
                return FALSE
            end if
            Revert simulation with one step back
        end while
        Continue simulation of $M(x, w)$ using $z$ and reach any halting state
    end for
    return TRUE
end procedure
```

The cannotReach subroutine visits c different nodes of $\mathcal{G}_{M,x,w}$ in order of ascending hash value, and for each of them checks that none of their neighbors hash to v. Since a single step of a computation only makes a local change, it is possible to remember this step and revert it afterward to continue with the next neighbor. If one of the neighbors hash to v or if a wrong non-deterministic guess has been made somewhere, we restore the aux-tape and return FALSE. Otherwise finish the simulation of M until a halting configuration is reached, to restore the orginal value of w. If we have visited c distinct nodes without finding v as a neighbor, then we return TRUE.

Property II follows easily from the correctness of the cannotReach subroutine: indeed, if $M(x, w)$ rejects, then acc_w is not reachable, and hence with the promise made on g and h_ℓ, some guess will cause cannotReach$(h_\ell(acc_w), M + 1, g, h_\ell)$ to return TRUE.

We now complete the proof of the theorem by proving Property I. Let us focus on the k-loop (lines 7–26) which goes through every hash function h_k. For each

h_k a value c is computed (see lines 8, 10, 13 and 20).

It might happen that the k-loop is aborted (in line 17), but if this never happens, then c will be compared to g (line 22), so that by the time the k-loop terminates, g will hold the maximum c produced for any value of k (line 23), and ℓ will hold the first value of k which produced this maximum (line 24).

Now we make the following two claims:

(i) If s is good, and h_k is one-to-one on $\mathcal{G}_{M,x,w}$, the i-loop (lines 9–21) will either abort, or set $c = |\mathcal{G}_{M,x,w}|$. Furthermore, some non-deterministic choice within the i-loop will not abort.

(ii) If s is good, but h_k is not one-to-one on $\mathcal{G}_{M,x,w}$, the i-loop will either abort, or set c to a value strictly smaller than $|\mathcal{G}_{M,x,w}|$. As above, some non-deterministic choice within the i-loop will not abort.

From these, it follows that if s is good, then for every k there is a non-deterministic guess which does not abort, and using any such non-aborting guess, g will be set to $|\mathcal{G}_{M,x,w}|$, and ℓ will be the smallest k for which h_k is one-to-one. This gives us Property I.

Let us prove claim (i). Suppose that h_k is one-to-one, and that the i-loop does not abort. Then we may prove inductively that in every iteration of the i-loop, c is the number of nodes in $\mathcal{G}_{M,x,w}$ reachable by $M(x, w)$ within $i - 1$ steps. Now, c, h_k satisfy the promise required by cannotReach, and hence, for any non-aborting guess, the v-loop will set c' to the number of nodes in $\mathcal{G}_{M,x,w}$ reachable within i steps; this value is then copied to c (line 20) for the next iteration of the i-loop. When the i-loop ends, c has been set to the number of nodes reachable within M steps, which is exactly $|\mathcal{G}_{M,x,w}|$. The fact that there always exists such a non-aborting guess follows from the behavior of the canReach procedure, and from the behavior of the cannotReach procedure in the case when c, h_k fulfill the promise.

To prove claim (ii), notice that the value of c' is incremented in line 13, and is thus bounded by the the size of image $h_k(\mathcal{G}_{M,x,w})$. So if h_k is not one-to-one, c' will always be strictly less than $|\mathcal{G}_{M,x,w}|$. On the other hand, it is always possible to find a non-deterministic guess which does not abort, even when h_k is not one-to-one. Whenever hash v is reachable in i steps, we can take the guess which makes canReach in line 12 return TRUE; when hash v is not reachable in i steps, we know from the behavior of cannotReach, that we can find a guess that makes cannotReach return true, provided that the argument c given to cannotReach in iteration i is not more than the number of nodes reachable within $i - 1$ steps. This follows from the fact that, in iteration $i - 1$, c' is bounded by the number of such nodes (because it is incremented only conditional on canReach of line 12. □

7.5 Hierarchies for Catalytic Computation

In this section we prove space-hierarchy theorems for deterministic and nondeterministic catalytic computation. Hierarchy theorems are usually proven using diagonalization. Since catalytic computation is a semantic model we do not know how to use diagonalization directly. Similarly to other semantic models (such as bounded-error randomized computation) we have to settle for hierarchy theorems with advice. This advice is used to tell the diagonalizing machine which machines can be safely simulated and diagonalized against, and which should not be simulated (so that the diagonalizing machine remains in the model).

The hierarchy theorem can be proven using the technique of Van Melkebeek and Pervyshev [vMP06], which are sophisticated variations of [Ž83]. Separations for certain space bounds follow directly from previous results on generic hierarchy theorems for semantic models of computation [KvM10, vMP06]. For some ranges of parameters we provide a direct proof, mainly the calculations justifying the correctness of the bounds. We state the theorem next.

7.5.1. Theorem. *Let* $a \geq 1$ *be an integer and* $s'(n)$ *and* $s(n)$ *be space-constructible functions. There is a function in the complexity class* $\mathsf{CNSPACE}(s(n))/1$ *that is not in* $\mathsf{CNSPACE}(s'(n))/a$*, and there is a function in* $\mathsf{CSPACE}(s(n))/1$ *that is not in* $\mathsf{CSPACE}(s'(n))/a$*, if any of the following is satisfied:*

1. $s'(n) = O(\log n)$ *and* $s(n) = \omega(\log n)$*.*
2. $s'(n) = O(\log^{k'} n)$ *and* $s(n) = \Omega(2^{(\log\log n)^{k'}})$*, for some constant* $k' > 1$*.*
3. $s'(n) = O(n^{k'})$ *and* $s(n) = \Omega(n^k)$*, where* $k, k' > 0$ *are reals such that* $k' < k/2$ *and* $k' < 1/(1+a)$*.*
4. $s'(n) = O(n^{k'})$ *and* $s(n) = \Omega(n^k)$*, where* $k, k' > 0$ *are reals such that* $k \geq 2a$ *and* $k \geq \lceil 4^{ak'^2} \rceil$*.*

Proof. The first part is immediate from Kinne and Van Melkebeek [KvM10] as catalytic computation satisfies the requirements on a *reasonable semantic model* and allows *complementation with linear-exponential overhead.*

Now we prove the third part using the technique of Van Melkebeek and Pervyshev [vMP06]. Fix a small enough $\epsilon > 0$ and let's consider the case when $s'(n) = n^{k'+\epsilon}$ and $s(n) = n^k$. Let M_i be an enumeration of possibly catalytic machines working in space $s'(n)$ with catalytic tape of size $2^{s'(n)}$. Assume without loss of generality that each machine appears infinitely often in this enumeration. We will construct a machine M and an advice sequence $\{b_n\}_{n>0}$ so that M/b_n behaves catalytically on inputs of length n and uses space at most $s(n)$ and catalytic space $2^{s(n)}$. No machine M_i will accept the same language as $M/\{b_n\}$ regardless of its a-bit advice.

The proof diagonalizes against all machines M_i with all possible advice sequences. We define a sequence of integers n_i and n_i^* as follows:

$$n_0^* = a,\ n_i = n_{i-1}^* + 1, \text{ and } n_i^* = n_i^{1+an_i}.$$

We will diagonalize against M_i with all possible advices on input of length between n_i and n_i^*. Let $m_i = \log n_i$ and for $j = 0, \dots, m_i$ define

$$n_{i,j} = n_i \cdot (n_i^a)^{2^j}.$$

For $w \in \{0,1\}^{a(m_i-j-1)}$ and $z \in \{0,1\}^a$ define $n_{i,j,wz} = n_{i,j} + \overline{wz}$, where $\overline{wz}$ is the integer represented by wz in binary. For $y \in \{0,1\}^{n_{i,j,wz}}$ define the function

$$f(y) = yz0^{n_{i,j+1,w}-n_{i,j,wz}-a}.$$

Since all $n_{i,j,wz}$ are distinct, this is a well defined partial function. We are ready to define the machine M which takes $\{b_n\}_{n>0}$ as its advice sequence.

1. On input x of length n do:
2. If $b_n = 0$ then REJECT.
3. If $n = n_{i,j,wz}$ for some i, j, wz, where $j \le m_i$, $|w| = a(m_i - j - 1)$ and $|z| = a$ then (nondeterministically) simulate M_i with advice z on input $f(x)$ and ACCEPT iff M_i accepts, and REJECT iff M_i rejects.
4. If $n = n_i^*$ then find y such that $f(f(\cdots f(y)\cdots)) = x$, where f is applied m_i-times. If no such y is found (such a y is a prefix of x) then REJECT. Let z be the first a bits of y. Using Savitch's algorithm decide whether M_i with advice z accepts y. If it accepts, REJECT, otherwise ACCEPT.

This defines the behavior of machine M. The advice $\{b_n\}_{n>0}$ is defined to be 1 of inputs of length $n_{i,j,wz}$ if and only if on all inputs of length $n_{i,j+1,w}$ machine M_i with advice z behaves in a correct catalytic manner (hence it is safe to simulate).

Assuming that machine M can perform the simulations in the designated space, it is easy to verify that it behaves catalytically and it diagonalizes against all machines M_i and all their possible advice sequences infinitely often.

So we only need to argue about the used space. Let M_i with advice sequence $\{z_n\}_{n>0}$ be a catalytic machine using work space $s'(n)$ and catalytic space $2^{s'(n)}$. On inputs of length $n_{i,j,wz_{n_{i,j+1,w}}}$, M will simulate M_i on inputs of length $n_{i,j+1,w}$ with advice $z_{n_{i,j+1,w}}$. By the choice of small enough ϵ, for all large enough n_i

$$\begin{aligned} s'(n_{i,j+1,w}) &\le 2n_i^{(1+2^{j+1}a)(k'+\epsilon)} \\ &\le n_i^{(1+2^j a)k} / n_i^{k/2} \quad \le \frac{s(n_{i,j,wz})}{n_i^{k/2}}. \end{aligned}$$

Hence, M can successfully simulate M_i on these input lengths using its work space and the catalytic space. It remains to verify that the space necessary for Savitch's algorithm on inputs of length n_i^* will fit into our work space. Savitch's algorithm for machine M_i on input y will require space at most $O((\log|y| + \log s'(|y|) + 2s'(|y|) + 2^{s'(|y|)})^2)$, which is less than $2^{3s'(|y|)}$ for y (resp. n_i) large enough. The length of y is at most $2n_i^{1+a}$. Thus

$$s'(|y|) \leq 2n_i^{(1+a)(k'+\epsilon)} < 2n_i$$

and

$$2^{3s'(|y|)} \leq 2^{6n_i} \leq s(n_i^*),$$

for n_i large enough.

To prove the second part one uses the same argument as above but verifies that the space needed by M for the simulations fits into its space bounds:

$$\begin{aligned}
s'(n_{i,j+1,w}) &\leq \left(\log 2n_i^{(1+2^{j+1}a)}\right)^{k'} \\
&= \left(1 + (1+2^{j+1}a)\cdot\log n_i\right)^{k'} \\
&\leq o\left(2^{\log^{k'}\left((1+2^j a)\cdot\log n_i\right)}\right) \\
&= o\left(2^{\log^{k'}\log n_i^{(1+2^j a)}}\right) \\
&= o(s(n_{i,j,wz})).
\end{aligned}$$

Similarly,

$$2^{3s'(2n_i^{1+a})} \leq o(s(n_i^*))\,.$$

For the fourth part we set the parameters exactly like Van Melkebeek and Pervyshev [vMP06, KvM10]: a constant $d = \max(2a, \lceil 4^{ak'^2}\rceil)$, $n_i^* = n_i^{n_i^d}$ and $n_{i,j} = n_i^{d^j}$. With these parameters there is sufficient space for M to simulate M_i's. □

7.A CNL definition, equivalent to Definition 7.3.2

7.A.1. DEFINITION. A decision problem L is in CNL if there exists a constant c and a deterministic Turing machine M, with a read-only input tape, a uni-directional certificate tape, work tape of size $c\log n$ and an auxiliary tape of size n^c, such that for all n-bit strings x and for all $w \in \{0,1\}^{n^c}$ it holds that

$$x \in L \iff \exists u \in \{0,1\}^{2^{n^c}}\ M(x,u,w) \text{ accepts}$$

and

$$\forall u \in \{0,1\}^{2^{n^c}}\ M(x,u,w) \text{ halts with } w \text{ on its aux-tape.}$$

The string u represents the contents of the uni-directional certificate tape, and w is the starting contents of the auxiliary tape.

Bibliography

[AAD00] M. Agrawal, E. Allender, and S. Datta. On TC0, AC0, and arithmetic circuits. *Journal of Computer and System Sciences*, 60(2):395–421, 2000.

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[ABK$^+$06] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM J. Comput.*, 35(6):1467–1493, June 2006.

[AG04] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

[AGSU15] A. Ambainis, W. Gasarch, A. Srinivasan, and A. Utis. Lower bounds on the deterministic and quantum communication complexity of Hamming-distance problems. *ACM Trans. Comput. Theory*, 7(3):10:1–10:10, June 2015.

[AK10] E. Allender and M. Koucký. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57(3), 2010.

[AMM14] M. Amy, D. Maslov, and M. Mosca. Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 33(10):1476–1489, Oct 2014.

[AMMR13] M. Amy, D. Maslov, M. Mosca, and M. Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 32(6):818–830, June 2013.

[AO94] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, pages 267–278, 1994.

[AS06] P. Arrighi and L. Salvail. Blind quantum computation. *International Journal of Quantum Information*, 4(05):883–898, 2006.

[BB84] C. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, volume 175, 1984.

[BBC$^+$93] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, 70(13):1895–1899, Mar 1993.

[BBD$^+$09] H. Briegel, D. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009.

[BBK$^+$13] J. Brody, H. Buhrman, M. Koucký, B. Loff, F. Speelman, and N. Vereshchagin. Towards a reverse newman's theorem in interactive information complexity. In *Computational Complexity (CCC), 2013 IEEE Conference on*, pages 24–33, June 2013.

[BBK$^+$16] J. Brody, H. Buhrman, M. Koucký, B. Loff, F. Speelman, and N. Vereshchagin. Towards a reverse newman's theorem in interactive information complexity. *Algorithmica*, pages 1–33, 2016.

[BBL$^+$15] J. Briët, H. Buhrman, D. Leung, T. Piovesan, and F. Speelman. Round elimination in exact communication complexity. In S. Beigi and R. König, editors, *10th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2015, May 20-22, 2015, Brussels, Belgium*, volume 44 of *LIPIcs*, pages 206–225. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[BC92] M. Ben-Or and R. Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992.

[BC94] S. Brands and D. Chaum. Distance-bounding protocols. In *EUROCRYPT'93*, pages 344–359. Springer, 1994.

[BCF$^+$11] H. Buhrman, N. Chandran, S. Fehr, R. Gelles, V. Goyal, R. Ostrovsky, and C. Schaffner. Position-based quantum cryptography: Impossibility and constructions. In P. Rogaway, editor, *Advances in Cryptology*

– *CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 429–446. Springer Berlin / Heidelberg, 2011.

[BCG$^+$16] H. Buhrman, Ł. Czekaj, A. Grudka, M. Horodecki, P. Horodecki, M. Markiewicz, F. Speelman, and S. Strelchuk. Quantum communication complexity advantage implies violation of a Bell inequality. *Proceedings of the National Academy of Sciences*, 113(12):3191–3196, March 2016.

[BCH86] P. Beame, S. Cook, and H. Hoover. Log depth circuits for division and related problems. *SIAM Journal on Computing*, 15(4):994–1003, 1986.

[BCK$^+$14] H. Buhrman, R. Cleve, M. Koucký, B. Loff, and F. Speelman. Computing with a full memory: Catalytic space. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 857–866, New York, NY, USA, 2014. ACM.

[BCL$^+$06] H. Buhrman, R. Cleve, M. Laurent, N. Linden, A. Schrijver, and F. Unger. New limits on fault-tolerant quantum computation. In *Foundations of Computer Science, 2006. FOCS '06. 47th Annual IEEE Symposium on*, pages 411–419, Oct 2006.

[BCP83] A. Borodin, S. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58(1–3):113–136, 1983.

[BCP$^+$13] J. E. Brody, S. Chen, P. A. Papakonstantinou, H. Song, and X. Sun. Space-bounded communication complexity. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 159–172, New York, NY, USA, 2013. ACM.

[BCW98] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC 1998)*, pages 63–68, 1998.

[Ben73] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 1973.

[BFK09] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 517–526. IEEE, 2009.

[BFS86] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 337–347, 1986.

[BFS92] J. Boyar, G. Frandsen, and C. Sturtivant. An arithmetic model of computation equivalent to threshold circuits. *Theoretical Computer Science*, 93(2):303–319, 1992.

[BFSS13] H. Buhrman, S. Fehr, C. Schaffner, and F. Speelman. The garden-hose model. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 145–158, New York, NY, USA, 2013. ACM.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the $\mathcal{P} =? \mathcal{NP}$ question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[BJ15] A. Broadbent and S. Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In R. Gennaro and M. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, volume 9216 of *Lecture Notes in Computer Science*, pages 609–629. Springer Berlin Heidelberg, 2015.

[BK11] S. Beigi and R. König. Simplified instantaneous non-local quantum computation with applications to position-based cryptography. *New Journal of Physics*, 13(9):093036, 2011.

[BKLS16] H. Buhrman, M. Koucký, B. Loff, and F. Speelman. Catalytic Space: Non-determinism and Hierarchy. In N. Ollinger and H. Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, volume 47 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:13, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Bro15a] A. Broadbent. Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, 2015.

[Bro15b] A. Broadbent. Popescu–Rohrlich correlations imply efficient instantaneous nonlocal quantum computation. *arXiv preprint arXiv:1512.04930*, 2015.

[BTV01] H. Buhrman, J. Tromp, and P. Vitányi. Time and space bounds for reversible simulation. In *Proceedings of the 28th ICALP*, 2001.

[Bus04] L. Bussard. *Trust Establishment Protocols for Communicating Devices.* PhD thesis, Eurecom-ENST, 2004.

[CB97] R. Cleve and H. Buhrman. Substituting quantum entanglement for communication. *Physical Review A*, 56(2):1201, 1997.

[CCJP10] S. R. Clark, A. J. Connor, D. Jaksch, and S. Popescu. Entanglement consumption of instantaneous nonlocal quantum measurements. *New Journal of Physics*, 12(8):083034, 2010.

[CCS06] S. Capkun, M. Cagalj, and M. Srivastava. Secure localization with hidden and mobile base stations. In *IEEE INFOCOM*, 2006.

[CDNT98] R. Cleve, W. v. Dam, M. Nielsen, and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Selected papers from the First NASA International Conference on Quantum Computing and Quantum Communications*, QCQC '98, pages 61–74. Springer-Verlag, 1998.

[CG75] D. Coppersmith and E. Grossman. Generators for certain alternating groups with applications to cryptography. *SIAM Journal on Applied Mathematics*, 29(4):624–627, 1975.

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, April 1988.

[CGMO09] N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky. Position based cryptography. In *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 391–407. Springer, 2009.

[CH05] S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *IEEE INFOCOM*, pages 1917–1928, 2005.

[Chi05] A. M. Childs. Secure assisted quantum computation. *Quantum Information & Computation*, 5(6):456–466, 2005.

[CL15] K. Chakraborty and A. Leverrier. Practical position-based quantum cryptography. *Phys. Rev. A*, 92:052304, Nov 2015.

[Cle89] R. Cleve. *Methodologies for Designing Block Ciphers and Cryptographic Protocols*. PhD thesis, University of Toronto, 1989.

[CM87] S. A. Cook and P. McKenzie. Problems complete for deterministic logarithmic space. *Journal of Algorithms*, 8(3):385–394, September 1987.

[Coo71] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[Coo85] S. A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64:2–22, 1985.

[CP10] A. Chattopadhyay and T. Pitassi. The story of set disjointness. *ACM SIGACT News*, 41(3):59–85, 2010.

[CSWX14] W. Y. Chiu, M. Szegedy, C. Wang, and Y. Xu. The garden hose complexity for the equality function. In Q. Gu, P. Hell, and B. Yang, editors, *Algorithmic Aspects in Information and Management*, volume 8546 of *Lecture Notes in Computer Science*, pages 112–123. Springer International Publishing, 2014.

[Dam91] C. Damm. DET=$\mathrm{L}^{(\#\mathrm{L})}$. Technical Report Informatik-Preprint 8, Fachbereich Informatik der Humboldt–Universität zu Berlin, 1991.

[DLTW08] A. C. Doherty, Y.-C. Liang, B. Toner, and S. Wehner. The quantum moment problem and bounds on entangled multi-prover games. In *Computational Complexity, 2008. CCC'08. 23rd Annual IEEE Conference on*, pages 199–210. IEEE, 2008.

[DNS10] F. Dupuis, J. B. Nielsen, and L. Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In *CRYPTO*, pages 685–706, September 2010.

[DSS16] Y. Dulek, C. Schaffner, and F. Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Advances in Cryptology – CRYPTO 2016*, pages 3–32. Springer Berlin Heidelberg, 2016.

[FBS$^+$14] K. Fisher, A. Broadbent, L. Shalm, Z. Yan, J. Lavoie, R. Prevedel, T. Jennewein, and K. Resch. Quantum computing on encrypted data. *Nature communications*, 5, 2014.

[FGKM15] S. Forest, D. Gosset, V. Kliuchnikov, and D. McKinnon. Exact synthesis of single-qubit unitaries over Clifford-cyclotomic gate sets. *Journal of Mathematical Physics*, 56(8):–, 2015.

[GC99] D. Gottesman and I. L. Chuang. Quantum Teleportation is a Universal Computational Primitive. *Nature*, 402:390–393, August 1999.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

[GKK$^+$07] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf. Exponential separations for one-way quantum communication complexity, with applications to cryptography. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 516–525, New York, NY, USA, 2007. ACM.

[Got98a] D. Gottesman. The Heisenberg representation of quantum computers. In *Group theoretical methods in physics. Proceedings, 22nd International Colloquium, Group22, ICGTMP'98, Hobart, Australia, July 13-17, 1998*, 1998.

[Got98b] D. Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57:127–137, Jan 1998.

[GS13] B. Giles and P. Selinger. Exact synthesis of multiqubit Clifford+T circuits. *Physical Review A*, 87(3):032332, 2013.

[HAM02] W. Hesse, E. Allender, and D. A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65(4):695–716, 2002.

[IH08] S. Ishizaka and T. Hiroshima. Asymptotic teleportation scheme as a universal programmable quantum processor. *Phys. Rev. Lett.*, 101(24):240501, Dec 2008.

[IH09] S. Ishizaka and T. Hiroshima. Quantum teleportation scheme by selecting one of multiple output ports. *Phys. Rev. A*, 79(4):042306, Apr 2009.

[IL95] N. Immerman and S. Landau. The complexity of iterated multiplication. *Information and Computation*, 116(1):103–116, 1995.

[Imm88] N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17(5):935–938, 1988.

[IP99] R. Impagliazzo and R. Paturi. The complexity of k-SAT. In *Proceedings of the 14th CCC*, pages 237–240, 1999.

[IW97] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 220–229, New York, NY, USA, 1997. ACM.

[JKMW09] R. Jozsa, B. Kraus, A. Miyake, and J. Watrous. Matchgate and space-bounded quantum computations are equivalent. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20090433. The Royal Society, 2009.

[JMRW15] N. Johnston, R. Mittal, V. Russo, and J. Watrous. Extended nonlocal games and monogamy-of-entanglement games. *arXiv preprint arXiv:1510.02083*, 2015.

[Kar72] R. M. Karp. *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972*, chapter Reducibility among Combinatorial Problems, pages 85–103. Springer US, Boston, MA, 1972.

[KL82] R. Karp and R. Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28:191–209, 1982.

[KMM13] V. Kliuchnikov, D. Maslov, and M. Mosca. Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates. *Quantum Info. Comput.*, 13(7-8):607–630, July 2013.

[KMS11] A. Kent, W. J. Munro, and T. P. Spiller. Quantum tagging: Authenticating location via quantum information and relativistic signaling constraints. *Phys. Rev. A*, 84:012326, Jul 2011.

[KMSB06] A. Kent, W. Munro, T. Spiller, and R. Beausoleil. Tagging systems, 2006. US patent nr 2006/0022832.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[KP14] H. Klauck and S. Podder. New bounds for the garden-hose model. In V. Raman and S. P. Suresh, editors, *34th International Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS 2014)*, volume 29 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 481–492, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[KvM02] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002.

[KvM10] J. Kinne and D. van Melkebeek. Space hierarchy results for randomized and other semantic models. *Computational Complexity*, 19(3):423–475, 2010.

[Lan88] L. J. Landau. Empirical two-point correlation functions. *Foundations of Physics*, 18(4):449–460, 1988.

[Lev73] L. A. Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

[LKB$^+$13] T. Lunghi, J. Kaniewski, F. Bussières, R. Houlmann, M. Tomamichel, A. Kent, N. Gisin, S. Wehner, and H. Zbinden. Experimental bit commitment based on quantum communication and special relativity. *Phys. Rev. Lett.*, 111:180504, Nov 2013.

[LL11] H.-K. Lau and H.-K. Lo. Insecurity of position-based quantum-cryptography protocols against entanglement attacks. *Phys. Rev. A*, 83(1):012322, Jan 2011.

[LMT97] K.-J. Lange, P. McKenzie, and A. Tapp. Reversible space equals deterministic space. In *Proceedings of the 12th Twelfth Annual IEEE Conference on Computational Complexity.*, 1997.

[LS07] T. Lee and A. Shraibman. Lower bounds in communication complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–399, 2007.

[Mal10a] R. A. Malaney. Location-dependent communications using quantum entanglement. *Phys. Rev. A*, 81(4):042319, Apr 2010.

[Mal10b] R. A. Malaney. Quantum location verification in noisy channels. In *GLOBECOM'10*, pages 1–6, 2010. arXiv:1004.4689v1.

[Mar14] O. Margalit. On the riddle of coding equality function in the garden hose model. In *Information Theory and Applications Workshop (ITA), 2014*, pages 1–5, Feb 2014.

[MM12] O. Margalit and A. Matsliah. Mage - the CDCL SAT solver developed and used by IBM for formal verification `http://ibm.co/P7qNpC`. personal communication, 2012.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge university press, 2000.

[New91] I. Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67 – 71, 1991.

[NPA08] M. Navascués, S. Pironio, and A. Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New Journal of Physics*, 10(7):073013, 2008.

[NRS01] G. Nebe, E. Rains, and N. Sloane. The invariants of the Clifford groups. *Designs, Codes and Cryptography*, 24(1):99–122, 2001.

[NV15] M. Navascués and T. Vértesi. Bounding the Set of Finite Dimensional Quantum Correlations. *Physical Review Letters*, 115(2):020501, July 2015.

[Pie11] K. Pietrzak. Personal communication, 2011.

[QLL$^+$15] B. Qi, H.-K. Lo, C. C. W. Lim, G. Siopsis, E. A. Chitambar, R. Pooser, P. G. Evans, and W. Grice. Free-space reconfigurable quantum key distribution network. *arXiv preprint arXiv:1510.04891*, 2015.

[QS15] B. Qi and G. Siopsis. Loss-tolerant position-based quantum cryptography. *Phys. Rev. A*, 91:042337, Apr 2015.

[Raz03] A. A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya Mathematics*, 67(1):145–159, 2003.

[Raz11] A. A. Razborov. Communication complexity. In D. Schleicher and M. Lackmann, editors, *An Invitation to Mathematics*, pages 97–117. Springer, 2011.

[RG15] J. Ribeiro and F. Grosshans. A tight lower bound for the BB84-states quantum-position-verification protocol. *arXiv preprint arXiv:1504.07171*, 2015.

[RS14] N. J. Ross and P. Selinger. Optimal ancilla-free Clifford+T approximation of z-rotations. *arXiv preprint arXiv:1403.2975*, 2014.

[RT92] J. Reif and S. Tate. On threshold circuits and polynomial computation. *SIAM Journal on Computing*, 21(5):896–908, 1992.

[Sav70] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177 – 192, 1970.

[Sel13] P. Selinger. Quantum circuits of T-depth one. *Physical Review A*, 87(4):042302, 2013.

[Sel15] P. Selinger. Efficient Clifford+T approximation of single-qubit operators. *Quantum Information & Computation*, 15(1-2):159–180, January 2015.

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science - FOCS 1994*, pages 124–134. IEEE, 1994.

[SHO13] S. Strelchuk, M. Horodecki, and J. Oppenheim. Generalized teleportation and entanglement recycling. *Phys. Rev. Lett.*, 110:010505, Jan 2013.

[Son14] H. Song. *Space-Bounded Communication Complexity.* PhD thesis, Tsinghua University, 2014.

[SP05] D. Singelee and B. Preneel. Location verification using secure distance bounding protocols. In *IEEE MASS'10*, 2005.

[Spe11] F. Speelman. Position-based quantum cryptography and the garden-hose game. Master's thesis, University of Amsterdam, 2011. arxiv:1210.4353.

[Spe16] F. Speelman. Instantaneous non-local computation of low T-depth quantum circuits. In *11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, volume 61 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:24, 2016.

[SSW03] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *WiSe'03*, pages 1–10, 2003.

[Sud78] I. H. Sudborough. On the tape complexity of deterministic context-free languages. *Journal of the ACM*, 25(3):405–414, July 1978.

[Sze88] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26(3):279–284, 1988.

[Sze12] M. Szegedy. Personal communication, 2012.

[TFKW13] M. Tomamichel, S. Fehr, J. Kaniewski, and S. Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, 2013.

[Tod91] S. Toda. Counting problems computationally equivalent to computing the determinant. *Technical Report CSIM*, 91-07, 1991.

[Tod92] S. Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Transactions on Information and Systems*, E75-D:116–124, 1992.

[Tur36] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1936.

[Unr14] D. Unruh. Quantum position verification in the random oracle model. In J. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, volume 8617 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2014.

[Vai03] L. Vaidman. Instantaneous measurement of nonlocal variables. *Phys. Rev. Lett.*, 90(1):010402, Jan 2003.

[Val79] L. G. Valiant. Completeness classes in algebra. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, STOC '79, pages 249–261, New York, NY, USA, 1979. ACM.

[Val92] L. G. Valiant. Why is Boolean complexity theory difficult? In *Poceedings of the London Mathematical Society symposium on Boolean function complexity*, pages 84–94. Cambridge University Press, 1992.

[Ven91] H. Venkateswaran. Properties that characterize LOGCFL. *Journal of Computer and System Sciences*, 43(2):380–404, 1991.

[Vin91] V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 270–284, 1991.

[vMP06] D. van Melkebeek and K. Pervyshev. A generic time hierarchy for semantic models with one bit of advice. In *Computational Complexity, 2006. CCC 2006. Twenty-First Annual IEEE Conference on*, pages 14 pp.–144, 2006.

[VN04] A. Vora and M. Nesterenko. Secure location verification using radio broadcast. In *OPODIS'04*, pages 369–383, 2004.

[VSBR83] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM Journal on Computing*, 12(4):641–644, 1983.

[Ž83] S. Žák. A Turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327 – 333, 1983.

[Weh06] S. Wehner. Tsirelson bounds for generalized clauser-horne-shimony-holt inequalities. *Physical Review A*, 73(2):022110, 2006.

[Wit15] P. Wittek. Algorithm 950: Ncpol2Sdpa—sparse semidefinite programming relaxations for polynomial optimization problems of noncommuting variables. *ACM Trans. Math. Softw.*, 41(3):21:1–21:12, June 2015.

[Yao79] A. C.-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11th annual ACM symposium on Theory of computing (STOC 1979)*, pages 209–213, 1979.

[Yao93] A. C.-C. Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science (FOCS 1993)*, pages 352–361, 1993.

[YGC12] L. Yu, R. B. Griffiths, and S. M. Cohen. Fast protocols for local implementation of bipartite nonlocal unitaries. *Physical Review A*, 85(1):012304, 2012.

[Yu11] L. Yu. Fast controlled unitary protocols using group or quasigroup structures. *arXiv preprint arXiv:1112.0307*, 2011.

[ZCC08] B. Zeng, X. Chen, and I. L. Chuang. Semi-Clifford operations, structure of C_k hierarchy, and gate complexity for fault-tolerant quantum computation. *Physical Review A*, 77(4):042313, 2008.

[ZL70] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematics Surveys*, 256:83–124, 1970.

[ZLFW06] Y. Zhang, W. Liu, Y. Fang, and D. Wu. Secure localization and authentication in ultra-wideband sensor networks. *IEEE Journal on Selected Areas in Communications*, 24:829–835, 2006.

Index

AC, 109
alphabet, 23, 120
auxiliary tape, 120, 131, 134

Bell measurement, 6, 19, 32, 64
Bounded Retrieval Model, 3
bra, **15**

catalytic computation, 9, **107**
  catalytic Turing machine, **120**
  non-deterministic, **134**
  Turing machine, 132
catalytic log space, *see* CL
catalytic non-deterministic log space, *see* CNL
Cauchy–Schwarz inequality, 14
Chernoff bound, 14, 46
Chinese Remainder Theorem, 133
CL, 10, 108, **121**
Cleve–Buhrman model, **22**
Clifford group, 8, 63
Clifford hierarchy, 62, 73
CNL, 10, 132, 135, 139
CNSPACE, 135
coCNL, 10, 132, 139
communication complexity, 21, 30
  bounded-error quantum, 22
  deterministic, 36
  exact quantum, 22
  quantum, **22**, 48
  randomized, 50
complexity theory, 23, 108
computational basis, 15, 89
configuration, 43
CSPACE, 120

density matrix, 20
DET, 110
detection probability, 91
distance bounding, 2

entanglement, 5, 16, 48, 68
EPR pair, 16, 30, 60, 68
equality function, 37
exponential-time hypothesis, 125

GapL, 109
garden-hose model, 7, **28**, 61, 66
  garden-hose protocol, **34**
  quantum, 30, 48, 86
  randomized, 30, 46
Gottesman–Chuang hierarchy, *see* Clifford hierarchy

Hadamard
  Hadamard basis, 4, 17, 89
  Hadamard matrix, 17
Hamming distance, **13**, 41, 49
Hamming weight, **13**, 37
hash function, 10, 132
Hilbert space, 15

IMM, 110
Immerman–Szelepcsényi Theorem, 10, 130, 139
inner product function, 37, 38
input tape, 23, 43, 120, 147
INQC, *see* instantaneous non-local quantum computation
instantaneous non-local quantum computation, 8, 60, **68**
Interleaved Product protocol, 63, **81**

ket, **15**
Kolmogorov complexity, **125**

L, 10, **24**, 29, 42, 67, 108, 130
logarithmic space, *see* L
LOGCFL, 109
#LOGCFL, 109

majority function, 37
match gates, 86
measurement, 17
  completeness, 18
  projective, 18
mixed state, 20
monogamy of entanglement game, 9, 92, 96

NC, 109
Ncpol2sdpa, 100
Newman's Theorem, 22
NL, 108
no-cloning theorem, 3, 21
Noisy Perfect Matching, 50
non-deterministic computation, 10, 24, 131
NP, 9, **24**
NP hard, 24

operator norm, **15**, 81, 102
output tape, 23, 131, 134

P, 9, **23**, 109
Pauli matrix, 17, 63
photon loss, 9, 90
polynomial time, 23
position verification, 2, 31, 59, 89
POVM, 18, 97
PRG, *see* pseudorandom generator
private randomness, 22
pseudorandom generator, 10, 130, 140
public randomness, 22, 48
pure state, 20

$\mathrm{QPV}_{\mathrm{BB84}}$, 4, 89
$\mathrm{QPV}_{\mathrm{BB84\text{-}e}}$, 9, **92**
quantum circuit, 8, 61
quantum homomorphic encryption, 87
quantum position verification, 30
quantum state, 14
quantum tagging, 3
qubit, 15

$\mathcal{S}$, 20, 68
SAC, 109
Savitch's theorem, 125, 130, 133
semidefinite programming, 9, 92, 98
simultaneous message passing, 42, 57
Solovay–Kitaev theorem, 82
space complexity, 23, 108
space hierarchy, 11, 130, 145
special relativity theory, 8
speed of light, 6, 9, 90

T-count, 8, **65**, 70
T-depth, 8, **65**, 78
tape head, 23
TC^1, 108
teleportation, 5, **18**, 30, 63
  port-based, 3, 60
tensor product, 15
transition function, 23
Turing machine, 9, 23, 108, 120
  deterministic, 43
  non-deterministic, 134
  oblivious, 43
  probabilistic, 131
  reversible, 43

unitary transformation, 16

vector norm, **15**
VP, 109

work tape, 23, 121, 131

zero-error probabilistic polynomial time,
see ZPP
ZPP, 10, **24**, 109, 131

Abstract

In this thesis, we present several results along two different lines of research. The first part concerns the study of position-based quantum cryptography, a topic in quantum cryptography. In the second part we introduce a new notion of computation, catalytic computation, and study this new model within complexity theory.

Part I: Position-based quantum cryptography

By combining quantum mechanics with special relativity theory, new cryptographic tasks can be developed that use the causality constraints of relativity theory in a constructive way. Position-based cryptography is a type of cryptography that wants to use location as a credential, instead of (or in addition to) a secret key – for instance to create a protocol to send messages that can only be read at one specific location.

After earlier proposals, which used only classical information, were shown to be insecure, new schemes for position-based cryptography that used quantum information at first seemed promising. Recent results showed that all such schemes can be broken by attackers that use an exponential amount of quantum resources. This leaves the following question open: Is it possible to create a scheme which is secure under realistic assumptions?

If an attack on a scheme requires more entanglement than the number of particles in the universe, then it surely is secure. Therefore, limiting the attacker's entanglement is a natural step. Our results will all consider this distinction: schemes that can be attacked using little entanglement are insecure, while schemes for which a coalition of attackers needs large entangled states will be secure.

Chapter 3. The first chapter on this topic analyzes a family of position-verification schemes that combines a single qubit with classical information. We introduce a new tool to study these schemes, the *garden-hose model*. In this simple com-

binatorial model, two parties, Alice and Bob, share 'pipes' between them, and they want to compute a function by linking these pipes together with 'hoses'. By studying garden-hose complexity, we characterize a class of teleportation attacks on the family of schemes, and show a surprising relationship between their security and open problems in computational complexity theory. We prove several smaller results on the new model and additionally introduce natural variants: the randomized garden-hose model, where the players share a random string, and the quantum garden-hose model, where Alice and Bob have access to a pre-shared entangled quantum state.

Chapter 4. In the next chapter we continue our study of the use of quantum information in position verification, but now our attention turns to a different class of protocols: those that that can be written using a class of small quantum circuits, those with low T-gate complexity. We combine techniques from blind and delegated quantum computation with the new garden-hose model and construct new efficient attacks on these schemes. As an additional application, we present an efficient attack on the Interleaved Product protocol for position verification, recently introduced by Chakraborty and Leverrier.

Chapter 5. The final chapter on this topic looks at questions that are directly inspired by practical considerations. Positioning protocols will likely use photons as carriers of quantum information, possibly traveling in optical fiber. This is incompatible with current protocols in two ways: a significant fraction of photons are lost in transmission, and the speed of light in fiber is lower than in vacuum. Adapting protocols to deal with these problems opens them up to new attacks. We propose a new protocol for position verification that prevents these attacks and use semidefinite programming to show security of this protocol against attackers that do not share entanglement.

Part II: Catalytic computation

In the second part of this thesis, we study the notion of a *catalytic-space computation.* This is a computation that has a small amount of clean space available and is equipped with additional auxiliary space, with the caveat that the additional space is initially in an arbitrary, possibly incompressible, state and must be returned to this state when the computation is finished. The term 'catalytic' comes from chemistry, where it refers to a reactant which speeds up a chemical reaction but is not consumed – just like the extra space available to the computation.

Chapter 6. In this chapter, we show that the extra space adds a surprising amount of power to the model. To obtain this result, we study an algebraic model of computation, called Transparent Programs, a variant of straight-line programs.

Within these Transparent Programs, we can adapt a construction by Ben-Or and Cleve to show that it's possible to compute TC^1 circuits using only a logarithmic amount of clean space. Additionally, we present some complexity-theoretical limits on the power of catalytic computation, by showing that computations that use a logarithmic amount of clean memory, can be simulated probabilistically in polynomial time.

Chapter 7. We continue the study of catalytic computation by translating two foundational results on space-bounded computation to this new setting. First we extend the model to incorporate nondeterminism. The Immerman–Szelepcsényi Theorem is an important classic result in complexity theory that shows that the complexity class of problems solvable by nondeterministic log-space Turing machines is closed under complement. We show that non-deterministic catalytic space is also closed under complement, under standard derandomization assumptions. Finally, we present a hierarchy theorem – we show that adding more space enables the catalytic computation to solve strictly more problems.

Samenvatting

In dit proefschrift worden resultaten gepresenteerd in twee verschillende onderzoeksrichtingen. Het eerste gedeelte heeft betrekking op *positionele quantum cryptografie*, een onderwerp binnen de quantum cryptografie. In het tweede gedeelte introduceren we een nieuw rekenmodel, *katalytische berekeningen*, en bestuderen dit model binnen de computationele complexiteitstheorie.

Deel I: Positionele Quantum cryptografie

Door quantummechanica te combineren met de speciale relativiteitstheorie kunnen nieuwe cryptografische taken worden bedacht die de beperkingen van relativiteitstheorie op een constructieve manier gebruiken. Positionele cryptografie is een type cryptografie waarbij de locatie gebruikt wordt als bewijs van de identiteit van een gebruiker, in plaats van (of samen met) een geheime sleutel. Een mogelijke toepassing is bijvoorbeeld een protocol voor het sturen van berichten die maar op één locatie gelezen kunnen worden.

Nadat de eerste voorstellen voor positionele cryptografie, gebaseerd op klassieke informatie, onveilig bleken te zijn, leken nieuwe protocollen gebaseerd op quantum informatie op het eerste gezicht veelbelovend. Recente resultaten lieten echter zien dat ook die protocollen onveilig zijn tegen aanvallers die exponentieel grote quantum toestanden kunnen manipuleren. Om deze reden richten wij ons op de vraag: Is het mogelijk om een schema te maken wat met realistische aannames veilig is?

Als een aanval op een cryptografisch protocol altijd extreem veel verstrengelde deeltjes nodig zou hebben, bijvoorbeeld meer dan het aantal deeltjes in het universum, dan zou dat protocol uiteraard veilig zijn tegen alle realistische aanvallen. Onze resultaten zullen allemaal dit onderscheid maken: protocollen die aangevallen kunnen worden met behulp van kleine verstrengelde toestanden zijn onveilig, terwijl we protocollen waarbij de aanvallers grote toestanden nodig hebben veilig noemen.

Hoofdstuk 3. Het eerste hoofdstuk over dit onderwerp analyseert positie-verificatie schema's die een enkele qubit combineren met klassieke informatie. We introduceren een nieuw hulpmiddel om deze schema's te bestuderen, het *garden-hose model*. In dit simpele combinatorische model delen twee partijen, Alice and Bob, 'buizen' met elkaar, en willen ze een functie berekenen door deze buizen met 'tuinslangen' aan elkaar vast te maken. Door garden-hose complexiteit te bestuderen, karakteriseren we een klasse van teleportatie-aanvallen op de schema's, en laten we een verrassende relatie zien tussen de veiligheid van deze schema's en open problemen in computationele complexiteitstheorie. We bewijzen diverse kleinere resultaten over het nieuwe model, en introduceren daarnaast twee varianten: het gerandomiseerde garden-hose model, waarbij de spelers een willekeurige reeks van bits delen, en het quantum garden-hose model, waarbij Alice en Bob toegang hebben tot een (vooraf gedeelde) quantum toestand.

Hoofdstuk 4. We vervolgen onze studie naar het gebruik van quantum informatie in positie verificatie, maar nu richten we onze aandacht op een andere klasse van protocollen: degene die beschreven kunnen worden door quantum circuits met lage T-gate complexiteit. We combineren technieken die eerder gebruikt zijn voor het delegeren van quantum berekeningen met het garden-hose model, en construeren zo nieuwe aanvallen op deze schema's. Als aanvullende toepassing presenteren we een efficiënte aanval op het Interleaved Product protocol voor positie verificatie, geïntroduceerd door Chakraborty en Leverrier.

Hoofdstuk 5. Ons laatste hoofdstuk over dit onderwerp beschouwt vragen die direct geïnspireerd zijn door praktische overwegingen. Positioneringsprotocollen zullen hoogstwaarschijnlijk fotonen gebruiken als dragers voor quantum informatie, mogelijk in glasvezelkabels. Dit komt niet overeen met de aannames die gedaan worden bij huidige protocollen op de volgende twee manieren: a significant aantal fotonen raakt verloren in overdracht, en de lichtsnelheid in glasvezel is lager dan de lichtsnelheid in vacuüm. Aanpassingen van protocollen om deze problemen op te lossen, geeft nieuwe aanvalsmogelijkheden en maakt de protocollen potentieel onveilig. We ontwikkelen een nieuw protocol voor positie verificatie dat deze aanvallen voorkomt, en we gebruiken semidefiniete programmering om te bewijzen dat dit protocol veilig is tegen aanvallers die geen verstrengelde toestand delen.

Deel II: Katalytische berekeningen

In het tweede gedeelte van dit proefschrift bestuderen we *berekeningen met katalytisch geheugen*. Dit zijn berekeningen die een kleine hoeveelheid lege geheugenruimte hebben en daarnaast uitgerust zijn met aanvullend hulpgeheugen, met het voorbehoud dat het aanvullende geheugen geïnitialiseerd is in een arbitraire, mogelijk niet-comprimeerbare, toestand en aan het eind van de berekening teruggezet moet zijn naar deze toestand. De term ‘katalytisch’ komt uit de scheikunde, waar het verwijst naar een reactant die een chemische reactie versnelt zonder verbruikt te worden – net als het extra geheugen wat bij de berekening beschikbaar is.

Hoofdstuk 6. In dit hoofdstuk laten we zien dat het extra geheugen een verrassende hoeveelheid mogelijkheden aan het rekenmodel toevoegt. Om dit resultaat te verkrijgen bestuderen we een algebraïsch rekenmodel, Transparante Programma’s. Met behulp van deze transparante programma’s kunnen we een constructie van Ben-Or en Cleve aanpassen en zo laten zien dat het mogelijke is om TC^1 circuits te simuleren met slechts een logaritmische hoeveelheid leeg geheugen. Daarnaast laten we enkele complexiteits-theoretische beperkingen zien van de kracht van katalytische berekeningen, door te bewijzen dat elke katalytische berekening die slechts een logaritmische hoeveelheid leeg geheugen gebruikt, gesimuleerd kan worden door een normale Turing machine in polynomiale tijd.

Hoofdstuk 7. We vervolgen de studie van katalytische berekeningen met het vertalen van twee klassieke resultaten uit de computationele complexiteitstheorie naar dit nieuwe model. Ten eerste breiden we het model uit met een non-deterministische variant. De Immerman–Szelepcsényi stelling is een belangrijk complexiteitstheoretisch resultaat dat laat zien dat de klasse van problemen die non-deterministische Turing machines kunnen oplossen met logaritmisch groot geheugen gesloten is onder complement. We laten zien dat non-deterministische katalytische berekeningen ook gesloten zijn onder complement, onder standaard derandomisatie-aannames. Tot slot presenteren we hiërarchie-stelling – we laten zien dat het toevoegen van meer geheugen de katalytische berekening strikt meer problemen laat oplossen.

Titles in the ILLC Dissertation Series:

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder. . . ": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**
Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**
Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**
Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**
Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**
Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**
Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**
Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**
Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2010-12: **Martin Mose Bentzen**
Stit, Iit, and Deontic Logic for Action Types

ILLC DS-2011-01: **Wouter M. Koolen**
Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
Small steps in dynamics of information

ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access

ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities

ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure

ILLC DS-2011-07: **Jop Briët**
Grothendieck Inequalities, Nonlocal Games and Optimization

ILLC DS-2011-08: **Stefan Minica**
Dynamic Logic of Questions

ILLC DS-2011-09: **Raul Andres Leal**
Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications

ILLC DS-2011-10: **Lena Kurzen**
Complexity in Interaction

ILLC DS-2011-11: **Gideon Borensztajn**
The neural basis of structure in language

ILLC DS-2012-01: **Federico Sangati**
Decomposing and Regenerating Syntactic Trees

ILLC DS-2012-02: **Markos Mylonakis**
Learning the Latent Structure of Translation

ILLC DS-2012-03: **Edgar José Andrade Lotero**
Models of Language: Towards a practice-based account of information in natural language

ILLC DS-2012-04: **Yurii Khomskii**
Regularity Properties and Definability in the Real Number Continuum: idealized forcing, polarized partitions, Hausdorff gaps and mad families in the projective hierarchy.

ILLC DS-2012-05: **David García Soriano**
Query-Efficient Computation in Property Testing and Learning Theory

ILLC DS-2012-06: **Dimitris Gakis**
Contextual Metaphilosophy - The Case of Wittgenstein

ILLC DS-2012-07: **Pietro Galliani**
The Dynamics of Imperfect Information

ILLC DS-2012-08: **Umberto Grandi**
Binary Aggregation with Integrity Constraints

ILLC DS-2012-09: **Wesley Halcrow Holliday**
Knowing What Follows: Epistemic Closure and Epistemic Logic

ILLC DS-2012-10: **Jeremy Meyers**
Locations, Bodies, and Sets: A model theoretic investigation into nominalistic mereologies

ILLC DS-2012-11: **Floor Sietsma**
Logics of Communication and Knowledge

ILLC DS-2012-12: **Joris Dormans**
Engineering emergence: applied theory for game design

ILLC DS-2013-01: **Simon Pauw**
Size Matters: Grounding Quantifiers in Spatial Perception

ILLC DS-2013-02: **Virginie Fiutek**
Playing with Knowledge and Belief

ILLC DS-2013-03: **Giannicola Scarpa**
Quantum entanglement in non-local games, graph parameters and zero-error information theory

ILLC DS-2014-01: **Machiel Keestra**
Sculpting the Space of Actions. Explaining Human Action by Integrating Intentions and Mechanisms

ILLC DS-2014-02: **Thomas Icard**
The Algorithmic Mind: A Study of Inference in Action

ILLC DS-2014-03: **Harald A. Bastiaanse**
Very, Many, Small, Penguins

ILLC DS-2014-04: **Ben Rodenhäuser**
A Matter of Trust: Dynamic Attitudes in Epistemic Logic

ILLC DS-2015-01: **María Inés Crespo**
Affecting Meaning. Subjectivity and evaluativity in gradable adjectives.

ILLC DS-2015-02: **Mathias Winther Madsen**
The Kid, the Clerk, and the Gambler - Critical Studies in Statistics and Cognitive Science

ILLC DS-2015-03: **Shengyang Zhong**
Orthogonality and Quantum Geometry: Towards a Relational Reconstruction of Quantum Theory

ILLC DS-2015-04: **Sumit Sourabh**
Correspondence and Canonicity in Non-Classical Logic

ILLC DS-2015-05: **Facundo Carreiro**
Fragments of Fixpoint Logics: Automata and Expressiveness

ILLC DS-2016-01: **Ivano A. Ciardelli**
Questions in Logic

ILLC DS-2016-02: **Zoé Christoff**
Dynamic Logics of Networks: Information Flow and the Spread of Opinion

ILLC DS-2016-03: **Fleur Leonie Bouwer**
What do we need to hear a beat? The influence of attention, musical abilities, and accents on the perception of metrical rhythm

ILLC DS-2016-04: **Johannes Marti**
Interpreting Linguistic Behavior with Possible World Models

ILLC DS-2016-05: **Phong Lê**
Learning Vector Representations for Sentences - The Recursive Deep Learning Approach

ILLC DS-2016-06: **Gideon Maillette de Buy Wenniger**
Aligning the Foundations of Hierarchical Statistical Machine Translation

ILLC DS-2016-07: **Andreas van Cranenburgh**
Rich Statistical Parsing and Literary Language

ILLC DS-2016-08: **Florian Speelman**
Position-based Quantum Cryptography and Catalytic Computation

ILLC DS-2016-09: **Teresa Piovesan**
Quantum entanglement: insights via graph parameters and conic optimization