LATVIJAS UNIVERSITĀTE

MAKSIMS KRAVCEVS

KVANTU UN VARBŪTISKO AUTOMĀTU DARBA SPĒJAS

Promocijas darbs

Rīga - 2006

LATVIJAS UNIVERSITĀTE
Matemātikas un informātikas institūts

MAKSIMS KRAVCEVS

KVANTU UN VARBŪTISKO AUTOMĀTU DARBA SPĒJAS

Promocijas darbs
datorzinātņu doktora (Dr. sc. comp.) zinātniskā grāda iegūšanai

Nozare: datorzinātnes
Apakšnozare: datorzinātņu matemātiskie pamati

Zinātniskais vadītājs:
Dr. habil. math., profesors
Rūsiņš - Mārtiņš Freivalds

Rīga - 2006

UNIVERSITY OF LATVIA
Institute of Mathematics and Computer Science

MAKSIMS KRAVCEVS

COMPUTATIONAL POWER OF QUANTUM AND PROBABILISTIC AUTOMATA.

Doctoral Thesis

Area: Computer Science
Sub-Area: Mathematical Foundations of Computer Science

Scientific Advisor:
Dr. habil. math., Professor
Rūsiņš Freivalds

Riga - 2006

Abstract

The thesis assembles research on two models of automata - probabilistic reversible (PRA) that appear very similar to 1-way quantum finite automata (1-QFA) and quantum one-way one counter automata (Q1CA), that is the most restricted model of non-finite space quantum automata. The objective of the research is to describe classes of languages recognizable by these models and compare related quantum and probabilistic automata.

We propose the model of probabilistic reversible automata. We study both one-way PRA with classical (1-C-PRA) and decide and halt (1-DH-PRA) acceptance. We show recognition of general class of languages $L_n = a_1^* a_2^* \dots a_n^*$ with probability $1 - \varepsilon$. We show whether the classes of languages they recognize are closed under boolean operations and describe general class of languages not recognizable by these automata in terms of "forbidden constructions" for the minimal deterministic automaton of the language. We also consider "weak" reversibility as equivalent definition for 1-way automata and show the difference from ordinary reversibility in 1.5-way case.

We propose the general notion of quantum one-way one counter automata(Q1CA). We describe well-formedness conditions for the Q1CA that ensure unitarity of its evolution. A special kind of Q1CA, called *simple*, that satisfies the well-formedness conditions is introduced. We show recognition of several non context free languages by Q1CA. We show that there is a language that can be recognized by quantum one-way one counter automaton, but not by the probabilistic one counter automaton.

Anotācija

Šis darbs apvieno pētījumus par diviem automātu veidiem: varbūtiskajiem apgriežamajiem automātiem (PRA), kas ir saistīti ar kvantu galīgajiem automātiem (QFA), un vienvirziena kvantu automātiem ar skaitītāju (Q1CA), kas ir ļoti ierobežots kvantu automātu modelis, kam atbilstoša kvantu sistēma nav galīga. Darba mērķis ir aprakstīt valodu klases, ko pazīst šie automāti, un salīdzināt kvantu un varbūtiskos automātus.

Mēs piedāvājam varbūtiskā apgriežama automāta modeli. Mēs pētām vienvirziena PRA gan ar klasisko (C-PRA) vārdu akceptēšanu, gan ar apstādināšanu (DH-PRA). Mēs parādām valodu klases $a_1^* a_2^* \ldots a_n^*$ pazīšanu ar PRA. Mēs parādām vai valodu klase, ko pazīst PRA, ir slēgta pret Būla operācijām. Mēs parādām vispārīgas valodu klases, ko C-PRA un DH-PRA nepazīst. Mēs apskatām vājas apgriežamības definīciju un parādām atšķirību no apgriežamības.

Mēs piedāvājam visparīgu kvantu vienvirziena automāta modeli ar skaitītāju (Q1CA). Mēs pierādām ka šis modelis apmierina transformācijas unitaritātes principu. Tiek piedāvāts speciāls Q1CA veids - vienkāršais Q1CA, kas ļauj konstruēt automātu piemērus konkrētām valodām. Mēs parādām vairāku kontekstatkarīgo valodu pazīšanu ar Q1CA. Mēs pierādām ka pastāv valodas, ko pazīst Q1CA, bet ko nepazīst varbūtiskais automāts ar skaitītāju.

Аннотация

Данная работа включает в себя исследование автоматов двух типов: вероятностных обратимых автоматов (PRA), которые связаны с конечными квантовыми автоматами (QFA), и однонаправленных квантовых автоматов со счетчиком (Q1CA), которые являются очень ограниченной моделью квантовых автоматов, для которых квантовая система не является конечной. Целью работы является описание классов языков, которые распознают эти автоматы, и сравнение квантовых и вероятностных автоматов.

Мы предлагаем модель вероятностного обратимого автомата. Мы изучаем однонаправленный PRA и с классическим распознаванием слов (C-PRA), и с распознаванием с остановкой (DH-PRA). Мы показываем, как PRA распознает класс языков $a_1^*a_2^* \dots a_n^*$. Мы выясняем является ли класс языков распознаваемых PRA замкнутым относительно Булевых операций. Мы показываем общие классы языков, которые не распознают C-PRA и DH-PRA. Мы вводим понятие слабой обратимости и показываем его отличие от понятия обратимости.

Мы предлагаем общую модель однонаправленного автомата со счетчиком (Q1CA). Мы доказываем, что эта модель обеспечивает унитарность трансформации. Предлагаем специальный вид Q1CA - простой Q1CA, который позволяет конструировать примеры автоматов для конкретных языков. Мы показываем, как Q1CA распознает некоторые контекстно-зависимые языки. Мы доказываем, что существуют языки распознаваемые Q1CA, которые не может распознать вероятностный автомат со счетчиком.

Preface

This thesis assembles the research performed by the author and reflected in the following publications:

1. M.Golovkins, M. Kravcevs, V. Kravcevs. On the Class of Languages Recognizable by Probabilistic Reversible Decide-and-Halt Automata. *iesniegts SOFSEM 2007 - 33rd Conference on Current Trends in Theory and Practice of Computer Science*, 10 lpp., 2007.

2. M.Golovkins, M. Kravcevs, V. Kravcevs. On the Class of Languages Recognizable by Probabilistic Reversible Decide-and-Halt Automata. Extended Abstract. *5th int. ERATO Conference on Quantum Information Systems. Proceedings, ERATO project*, pp. 131-132, 2005.

3. M.Kravcevs , Better Probabilities for One-Counter Quantum Automata. *6th International Baltic Conference on Data Bases and Information Systems. Proceedings, University of Latvia*, pp. 128-135, 2004.

4. M. Golovkins, M. Kravtsev. Probabilistic Reversible Automata and Quantum Automata. *COCOON 2002 Proceedings, Lecture Notes in Computer Science, Springer-Verlag*, Vol. 2387, pp. 574-583, 2002.

5. M. Golovkins, M. Kravtsev. Probabilistic Reversibility and Its Relation to Quantum Automata. *Quantum Computation and Learning. 3rd International Workshop. Proceedings, Malardalen University Press*, pp. 1-22, Riga, 2002.

6. R. Bonner, R.M. Freivalds. M. Kravcevs. Quantum versus Probabilistic One Counter Finite Automata. *SOFSEM 2001, Lecture Notes in Computer Science, Springer-Verlag*, Vol. 2234, pp. 181. - 190, 2001.

7. R. Bonner, R.M. Freivalds. M. Kravcevs. Quantum versus Probabilistic One Counter Finite Automata. Extended abstract, *Quantum Computation and Learning. 2nd International Workshop. Proceedings, Malardalen University Press*, pp. 80.-88, 2000.

8. M. Kravcevs Quantum One Counter Finite Automata. *SOFSEM'99: 26th Conference on Current Trends in Theory and practice of Informatics, Lecture Notes in Computer Science, Springer-Verlag*, Vol. 1725, pp. 431-440, 1999

The results of the thesis were presented at the following international conferences and workshops:

1. 5th ERATO Conference on Quantum Information Systems. Tokyo, 2005, August 24-31, Poster "On DH-Probabilistic reversible automata".

2. 6th International Baltic Conference on Data Bases and Information Systems, Riga, 2004, July 6-9, Presentation "Better Probabilities for Quantum One Counter automata".

3. 7th workshop on Quantum Information Processing, (QIP'2004), Waterloo University, Canada , 2004. January 14-19, Poster "Probabilistic Reversible Automata".

4. Computing and Combinatorics. 8th Annual International Conference, COCOON 2002, Singapore, August 15-17. Presentation "Probabilistic Reversible Automata and Quantum Automata". Co-presented by Marats Golovkins

5. Quantum Computation and Learning. 3rd International Workshop. Riga, Latvia, May 25-26, 2002. Presentation "Quantum Automata and Probabilistic Reversible Automata".

6. 5th workshop on Quantum Information Processing (QIP'2002), New York, IBM Watson Research Center, Jan 14-17, 2002. Poster "Quantum One Counter Automata".

7. 4th workshop on Quantum Information Processing, Amsterdam, Jan 19-22,2001. Poster on "Quantum Automata and Probabilistic Reversible Automata".

8. Quantum Computation and Learning. 2nd International Workshop. Sundbyholms Slott, Sweden, May 27-29, 2000. Presentation "Quantum One Counter Automata versus Probabilistic One Counter Automata".

9. SOFSEM'99: Theory and Practice of Informatics. 26th Conference on Current Trends in Theory and Practice of Informatics. Milovy, Czech Republic, November 27 - December 4. Presentation "Quantum One Counter Automata".

10. Quantum Computation and Learning. 1st International Workshop. Riga, Latvia, September 11-13, 1999. Presentation "Quantum One Counter Automata".

I thank my co-authors Richard Bonner, Rūsiņš Freivalds, Marats Golovkins, Vasilijs Kravcevs, who significantly contributed to this research. I thank Arnolds Ķikusts and Andrey Dubrovski for useful discussions.

Especially I thank my supervisor Prof. Rūsiņš Freivalds, whose ideas, support and positive attitude was one of the key factors for me to pursue and complete the research.

Contents

Chapter 1

Introduction

For those interesting in quantum computation in general we refer to the monographs of J. Gruska [Gr 99] and M.Nielsen and I.Chuang [NC 00] for the complete overview on the subject.

1.1 Background on quantum Turing machine

We refer to the [BV 97] for the description of quantum Turing Machine. There are two ways of thinking about quantum computers. One way that may appeal to computer scientists is to think of a quantum Turing Machine as a quantum physical analogue of a probabilistic Turing Machine. It has an infinite tape and a transition function, and actions of the machine are local and completely specified by this transition function. Unlike probabilistic Turing Machines, quantum Turing Machines allow branching with complex “probability amplitudes”, but impose the further requirement that the machine’s evolution be time - reversible. Another way is to view a quantum computer as effecting a transformation in a space of complex superpositions of configurations. Quantum physics requires that this transformation to be unitary. A quantum algorithm may then be regarded as the decomposition of a unitary transformation into a product of unitary transformations, each of which makes only simple local changes. A precise model of quantum computational device was formulated by Deutch [De 85], he proved that quantum Turing machines compute exactly the same recursive functions as classical deterministic Turing machines do. Yao [Y 93] extended this by proving that quantum circuits are polynomially equivalent to quantum Turing machines. Bernstein and Vazirani [BV 97] showed an efficient universal quantum Turing machine. They also considered relevant complexity classes and define quantum analogy of **BPP** [1] class - **BQP** [2] and prove that **BPP** $\subseteq$ **BQP** $\subseteq$ **PSPACE**, thus establishing that it will not be possible to conclusively prove that **BPP** $\neq$ **BQP** without resolving the major open problem **P** $\stackrel{?}{=}$ **PSPACE**. They actually prove stronger result **BQP** $\subseteq$ $\mathbf{P}^{\sharp\mathbf{P}}$.

They also gave the evidence that **BQP** $\neq$ **BPP**, by proving the existence of an oracle relative to which there are problems in **BQP** that cannot be solved with small error probability by probabilistic machines restricted to running in $n^{o(logn)}$ steps. Simon [Si 94] strengthened this evidence by proving the existence of an oracle relative to which **BQP** cannot even be simulated by probabilistic machines allowed to run for $2^{n/2}$ steps. In addition, Simon's paper also introduced a technique that was one of the components of the famous Shor's results [Sh 94]. These results shows that there are certain classes of problems where quantum devices can compute much more efficiently then classical ones.

It is natural to ask whether quantum Turing Machines can solve every problem in NP in polynomial time. Bennett, Bernstein, Brassard and Vazirani [BBBV 97] give evidence showing the limitations of quantum Turing Machines. They show that relative to an oracle chosen uniformly at random, with probability 1, the class NP cannot be solved on a quantum Turing Machine in time $o(2^{n/2})$. The bound is tight since work of Grover [G 97] shows how to accept the class NP relative to any oracle on a quantum computer in time $O(2^{n/2})$.

1.2 Background on quantum automata

1.2.1 Background on quantum finite automata

Quantum one way finite automaton can be considered as most restricted model of quantum computation. It describes the evolution of finite dimensional quantum system which dimension is independent on the length of computation. The quantum device is controlled by the classical part - that sequentially reads letters of the word from the input and applies certain

[1] **BPP** is the class of decision problems (languages) that can be solved in polynomial time by probabilistic Turing machines with error probability bounded by 1/3 (for all inputs). Using standard boosting techniques, the error probability can then be made exponentially small in respect to 1/k by iterating the algorithm k times and returning the majority answer.

[2] **BQP** is the class of decision problems (languages) that can be solved in polynomial time by quantum Turing machines with error probability bounded by 1/3, as is the case with **BPP**, the error probability of **BQP** machines can be made exponentially small [BBBV 97]

quantum operations dependant on the letter read on the quantum system. So one can speculate that such device is relatively simple to be built.

There are proposed different models of quantum finite automata(QFA). Common for all models is that for each letter of input alphabet there is defined a separate unitary transformation. Thus the computation on a word consists of consecutive application of the unitary transformations according to the letters of the word on the state of the underlying finite dimensional quantum system and possibly subsequent measurement. Major differences between models are what are the allowed measurements and definition of acceptance. Once we have measurements as intermediate steps we talk about quantum system with mixed states as it is with some probability in one of the possible states outcomes of the measurement.

There are commonly used 2 models of language acceptance for quantum automata. In *classical* acceptance model the states are divided into 2 disjoint sets of accepting and non-accepting automata and the automaton accepts the word, if it is in accepting state after having read the last symbol of the word and rejects otherwise. In *decide and halt* acceptance model the states are divided into 3 disjoint sets of accepting, rejecting and non-terminating states, and automaton after each step can halt, it accepts computation if it is in an accepting state, rejects in a rejecting state and continues computation otherwise.

One-way classical QFA with pure states commonly referred in literature as *measure-once* QFA (MO-QFA) were introduced by C. Moore and J. P. Crutchfield in [MC 97]. It is most straightforward definition of QFA as allows no measurement during computation. The only measurement is done after the word is processed to obtain classical result - whether word is accepted or rejected. C. Moore and J. P. Crutchfield showed the class of languages recognizable by these automata, to be group finite languages. In [BP 99] A. Brodsky and N. Pippenger noted that MO-QFA recognize the same language class as permutation automata.

Subsequently, A. Kondacs and J. Watrous introduced "decide and halt" 1-way QFA with pure states refered as *measure-many* QFA (MM-QFA) in [KW 97]. This definition differs from the previous one as allows a special measurement to be performed after unitary transformation when reading each letter. The measurement projects state of the automaton to one of three subspaces, one that corresponds to accepting states of automata one to the rejecting states and one to the non-halting states. The computation halts if accepting or rejecting state is observed and continues otherwise. Thus probabilistic decision is done on every letter and that gives additional power to the automata comparing with QFA-MO. We don't get mixed states in this model as computation is continued only with single quantum state.

In [KW 97] it is shown that anyway this model recognizes only the subset of the regular languages. A. Ambainis and R. Freivalds [AF 98] showed that the same class of languages as for DRA is recognizable with probability of correct answer over $\frac{7}{9}$, some other regular languages like a^*b^* can be recognized with smaller probabilities. There is a number of researches on this model, but still the class of languages recognizable by QFA-MM is not defined completely. In particular in [BP 99] A. Brodsky and N. Pippenger noted that MO-QFA recognize the same language class as permutation automata ([T 68]) and described class of languages not recognizable by MM-QFA by presenting "forbidden construction". If minimal deterministic automaton for the language contains such a construction then the language can not be recognized by QFA-MM. A. Ambainis, A. Ķikusts and M. Valdats determined in [AKV 00] that the class of languages recognized by MM-QFA is not closed under boolean operations, as well as significantly improved the necessary condition of a language to be recognized by MM-QFA, proposed by [BP 99], presenting a number of other "forbidden constructions". Still the exact class of languages for MM-QFA is not determined.

For some languages [AF 98] (example $L_p = \{a^i | i \text{ is divisible by } p\}$) the size of quantum automata can be exponentially smaller then in deterministic and probabilistic case. The opposite results for languages having deterministic automata with exponentially smaller size then quantum counterpart are also shown in [AF 98] and [ANTV 98]. An example from [ANTV 98] $L_n = \{wa | w \in \{a, b\}^*, |w| < n\}$ requires $2^{\Omega(n)}$ states in quantum case.

More general then QFA-MM model is offered by A. Nayak in [N 99] by allowing any *orthogonal* measurement as a valid intermediate computational step. Processing of each input letter in this model means applying of finite sequence of unitary transformations and *orthogonal* measurements, followed by the measurement according to the QFA-MM case on fixed subspaces formed by accepting, rejecting and non-halting states. This model may be seen as a finite memory version of the mixed state quantum computers defined in [AKN 97]. This model does not allow the more general positive operator valued measurements because the implementation of such measurements involves the joint unitary evolution of the state of the automaton with a fresh set of ancilla qubits, which runs against the fixed finite workspace spirit of the model.

The automata of that model can recognize only the subset of regular languages anyway. An example from [N 99] of regular language that is not recognizable by enhanced QFA is $\{a, b\}^*a$.

We can consider Nayak's enhanced QFA model with classical acceptance - with no measurements according to the QFA-MM rules, but with *orthogonal* measurement as a valid intermediate computational step combined with unitary transformation. It is similar counterpart for Nayak's model, as QFA-MO for QFA-MM. It is mixed state model with classical acceptance in our terminology.

This model is considered in the [ABGKMT 06]. In the paper it is referenced as "latvian QFA". Class of languages recognizable by this model is found, it is expressed in algebras terms. It has the properties of closure under boolean operations. This class can also be expressed in terms of forbidden constructions for the minimal deterministic automata.

Presented above are models connected with topic of research in Probabilistic Reversible Automata. There is a number of less widely used and spread models. They usually are used to illustrate certain quantum combined with classical computation effects. For example see [BMP 03] for model with control language, see [Dz 03] for model with arbitrary measurement allowed controllable by the state of automata. In the paper [BMP 03] there is shown that class of languages recognizable by QFA-MM coincides with the class for QFA with control language of particular form of this language, but still it does not give the description of that class.

1.2.2 Background on non finite dimensional quantum automata

There have been considered several models of quantum automata that are not longer finite dimensional quantum systems.

First of them 2-way QFA is considered by A.Kondacs and J.Watrous in [KW 97]. The automaton can decide on each step on the direction to move, thus configuration of automaton is not only state, but also position on the input tape. That means that quantum system realizing such automata should have basis corresponding to the automata configurations, and state of automaton is a vector in this space is. They prove that it can recognize any regular language. In fact recognition of all regular languages is proven already for deterministic version of this automaton. The deterministic reversible automaton can be seen as quantum automaton with one-to-one transition between configurations, thus its state always corresponds to single configuration not the superposition of them, and then also measurement does not affect the state.

There is shown that 2-way QFA can also recognize some nonregular languages in polynomial time. Example in [KW 97] is given on language $a^n b^n$ recognition (Probabilistic 2-way automata can do that in exponential time only, see [Fr 81, DSt 89]). The recognition of non regular languages is possible in 2-way QFA due to the effect of interference between "different" computational pathes and parallel operation over the whole word. In [AI 99] M. Amano and K. Iwama showed that emptiness problem is not decidable for 2-way (even 1.5-way) quantum finite automata. In [Du 01] it is actually determined that even 1.5-way QFA can recognize non-regular languages, which is not possible by deterministic and probabilistic counterparts of this model.

Regarding other models we should mention pushdown quantum automata by C. Moore and J. P. Crutchfield [MC 97], but most of their results is for generalized model that actually can not be considered truly "quantum". Another definition of pushdown automata was offered by M. Golovkins in [Go 00], he showed recognition of all regular languages by this model and several non context free languages. Quite comprehensive research on counter automata is done by T. Yamasaki, H. Kobayashi and H. Imai in [YKI 02]. They focus on 2-way automata and also on 2 counter automata. For these models it stands that they are more powerful then deterministic models, but still there is no clear comparison with probabilistic automaton.

1.2.3 Research and Results

Quantum 1-counter automata

We propose the notion of quantum 1 counter automata (Q1CA) in [K 99]. The simplest one-way model is considered. The main idea is to present a model with non-finite dimension of underlying quantum system, but still very restricted. In [N 99] there is shown that quantum automata in order to recognize forbidden construction for enhanced QFA $wa|w \in (a,b)*|w| < n$ should have at least $2^{\Omega(n)}$ quantum states in underlying quantum system, but Q1CA has only $card(Q) * n$, where Q is set of states of automaton.

We can consider one counter automata as a subclass of pushdown automata. The easiest approach is to consider pushdown automata with single letter alphabet of the stack, but there is a difference as counter is allowed to hold negative values [3].

In our research we describe well-formedness conditions for the Q1CA that ensure unitarity of its evolution. Also a special kind of Q1CA, called *simple*, that satisfies the well-formedness conditions is introduced. Transition matrix of simple automata is defined as a set of unitary matrixes for each input letter and separate for zero and non zero value of the counter. By these matrixes transformation of states is determined. But the change of the value of the

[3] so properly we should consider 2 symbols' alphabet of the stack, one for negative and one for positive values of the counter and forbid transitions placing into the stack the other type of symbol once stack is not empty

counter depends only on the state automaton moves in. That definition allows to construct concrete Q1CA.

We show recognition of several non context free languages by Q1CA. The languages are $L_1 = \{ \ 0^i 10^j 10^k \ \| \ \text{(i=k or j=k) and } \neg\text{(i=j))} \ \}$ and $L_2 = \{ \ 0^i 10^j 10^k \ \| \ \text{exactly 2 of i,j,k are equal}\}$. The [YKTI 00] show that these languages are also recognizable by probabilistic one counter automata. In quantum case we can achieve some better probabilities however [K 04].

The main result regarding one counter automata are received in cooperation with R. Freivalds and R. Bonner [BFK 01]. We show that there is a language that can be recognized by quantum one-way one counter automaton, but not by the probabilistic one counter automaton. That completes our research on quantum 1 counter automata.

Model of Probabilistic Reversible Automata

We propose the model of probabilistic reversible automata in [GK 02]. There is a number of reasons for inventing that model when observing QFA. First of all if we consider Nayak's model of quantum automata with mixed states [N 99], with measurements only on the single dimensional subspaces, we get a probabilistic automaton which transition matrixes are *doubly* stochastic [4]. Another reason is that in [AF 98] there is presented an example of automaton that makes a probabilistic decision on the first step and then behaves as deterministic reversible automaton. But restriction of having probabilistic split only on the first letter seems quite artificial. That explains our approach to definition of PRA: property of transition operator to be doubly stochastic is used to define probabilistic reversible automata. We should note as not every double stochastic matrix has an unitary prototype that we can not consider this model as subclass of quantum automata.

Additional objective to present such model was to be able to split probabilistic and quantum effects that arise in calculation by quantum automata. So we have studied properties of probabilistic reversible automata. Some of them appeared very similar to those of corresponding QFA, but some other not.

One-way C-PRA

We consider one-way PRA with classical acceptance. We prove that C-PRA recognize the class of languages $a_1^* a_2^* \ldots a_n^*$ with probability $1 - \varepsilon$. This class can be recognized by MM-QFA, with worse acceptance probabilities, however

[4] doubly stochastic means that the sum of elements in every column and row of transition matrix equals to 1

[ABFK 99]. This result also implies that Nayak's enhanced QFA recognize this class of languages with probability $1 - \varepsilon$.

We show that any language recognizable by C-PRA with probability 1 can be recognized by DRA. But in case a language is recognizable by C-PRA with bounded error we can boost this probability arbitrary close to 1, similarly as for any PRA.

That allows us to show that the class of languages recognized by C-PRA is closed under boolean operations, inverse homomorphisms and word quotient. But it is not closed under homomorphisms.

Further, we show general class of regular languages, not recognizable by C-PRA. We express this class as set of forbidden construction for minimal deterministic automata. In particular, such languages as (a,b)*a and a(a,b)* are in this class. This class has strong similarities with the class of languages, not recognizable by MM-QFA [AKV 00]. There are 2 forbidden constructions, first one is exactly the one as for MM-QFA [BP 99] the second one includes the one considered in [AKV 00].

We refer to the subsequent research by A.Ambainis at all [ABGKMT 06] which shows that the class of languages recognizable by C-PRA and C-QFA is all regular languages except languages containing these forbidden constructions.

One-way DH-PRA automata

Properties of DH-PRA model are studied in [GKK 05] and [GKK 06]. Obviously C-PRA recognize proper subset of languages recognizable by DH-PRA, for example C-PRA can not recognize a(a,b)* that can be recognized by DH-PRA.

We prove forbidden constructions for DH-PRA. These constructions are very similar to the constructions for MM-QFA considered in [AKV 00].

We are not able to show the exact class of languages recognizable by DH-PRA. The unknown gap is left for languages which minimal automaton contains forbidden construction for C-PRA but not for DH-PRA.

We show that the class of languages recognized by DH-PRA is not closed under union. This proof is similar to the proof for MM-QFA in [AKV 00].

As results regarding MM-QFA are quite similar that leads to a conjecture that class of languages recognizable by DH-PRA is likely to include the one recognizable by MM-QFA or these classes are equal. Still we are unable to prove or disprove that.

1.5 way C-PRA automata

There could be 2 different approaches to define probabilistic reversible automata. Another approach is to consider reverse of the transition function. We call automaton of some type "weakly reversible " if the reverse of its transition function corresponds to the transition function of a valid automaton of the same type. In the case of 1-way automata definitions are equivalent.

We see the difference in 1.5 way automata case, showing the recognition of $\{a, b\}^*a$ by weakly 1.5-way PRA. It is believed that that is impossible for the 1.5-way PRA.

Chapter 2

Preliminaries

In this chapter we consider notions, definitions and well-known or elementary facts, referenced directly or indirectly further in the thesis.

2.1 Unitary and Stochastic Operations

In this section, we recall well known definitions and theorems from linear algebra. We also consider elementary properties of Doubly Stochastic Matrixes. For the sake of completeness, some of the theorems are supplied with elementary proofs.

Unitary Matrixes

As noted in the next sections infinite unitary matrixes with finite number of nonzero elements in each row and column describe the work of quantum automata. Further lemmas state some properties of such matrixes.

Definition 2.1. *A complex matrix U is called* unitary, *if $UU^* = U^*U = I$.*

Lemma 2.2. *If matrixes A and B are unitary, then their direct product is a unitary matrix.*

If U is a finite matrix, then $UU^* = I$ iff $U^*U = I$. However this is not true for infinite matrixes:

Example 2.3.

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \dots \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Here $U^*U = I$ but $UU^* \neq I$.

Lemma 2.4. *If infinite matrixes A, B, C have finite number of nonzero elements in each row and column, then their multiplication is associative: $(AB)C = A(BC)$.*

Proof. The element of matrix $(AB)C$ in i-th row and j-th column is $k_{ij} = \sum\limits_{s=1}^{\infty}\sum\limits_{r=1}^{\infty} a_{ir}b_{rs}c_{sj}$. The element of matrix $A(BC)$ in the same row and column is $l_{ij} = \sum\limits_{r=1}^{\infty}\sum\limits_{s=1}^{\infty} a_{ir}b_{rs}c_{sj}$. As in the each row and column of matrixes A, B, C there is a finite number of nonzero elements, it is also finite in the given series. Therefore the elements of the series can be rearranged, and $k_{ij} = l_{ij}$. □

Lemma 2.5. *If $U^*U = I$ have finite number of nonzero elements in each row and column, then the norm of any row in the matrix U does not exceed 1.*

Proof. Let us consider the matrix $S = UU^*$. The element of this matrix $s_{ij} = \langle r_j | r_i \rangle$, where r_i is i-th row of the matrix U. Let us consider the matrix $T = S^2$. The diagonal element of this matrix is

$$t_{ii} = \sum_{k=1}^{\infty} s_{ik}s_{ki} = \sum_{k=1}^{\infty} \langle r_k | r_i \rangle \langle r_i | r_k \rangle = \sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2.$$

On the other hand, taking into account Lemma 2.4, we get that

$$T = S^2 = (UU^*)(UU^*) = U(U^*U)U^* = UU^* = S.$$

Therefore $t_{ii} = s_{ii} = \langle r_i | r_i \rangle$. It means that

$$\sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2 = \langle r_i | r_i \rangle. \tag{2.1}$$

This implies that every element of series (2.1) does not exceed $\langle r_i | r_i \rangle$. Hence $|\langle r_i | r_i \rangle|^2 = \langle r_i | r_i \rangle^2 \leq \langle r_i | r_i \rangle$. The last inequality implies that $0 \leq \langle r_i | r_i \rangle \leq 1$. Therefore $|r_i| \leq 1$. □

Lemma 2.6. *Let us assume that $U^*U = I$. Then the rows of the matrix U are orthogonal iff every row of the matrix has norm* 0 *or* 1*.*

Proof. Let us assume that the rows of the matrix U are orthogonal. Let us consider equation (2.1) from the proof of Lemma 2.5, i.e., $\sum_{k=1}^{\infty} |\langle r_k|r_i\rangle|^2 = \langle r_i|r_i\rangle$. As the rows of the matrix U are orthogonal, $\sum_{k=1}^{\infty} |\langle r_k|r_i\rangle|^2 = |\langle r_i|r_i\rangle|^2$. Hence $\langle r_i|r_i\rangle^2 = \langle r_i|r_i\rangle$, i.e., $\langle r_i|r_i\rangle = 0$ or $\langle r_i|r_i\rangle = 1$. Therefore $|r_i| = 0$ or $|r_i| = 1$.

Let as assume that every row of the matrix has norm 0 or 1. Then $\langle r_i|r_i\rangle^2 = \langle r_i|r_i\rangle$ and in compliance with the equation (2.1), $\sum_{k\in n^+\setminus\{i\}} |\langle r_k|r_i\rangle|^2 = 0$. This implies that $\forall k \neq i \; |\langle r_k|r_i\rangle| = 0$. Hence the rows of the matrix are orthogonal. □

Lemma 2.7. *The matrix U is unitary iff $U^*U = I$ and its rows have norm 1.*

Proof. Let us assume that the matrix U is unitary. Then in compliance with Definition 2.1, $U^*U = I$ and $UU^* = I$, i.e, the rows of the matrix are orthonormal.

Let us assume that $U^*U = I$ and the rows of the matrix are normalized. Then in compliance with Lemma 2.6 the rows of the matrix are orthogonal. Hence $UU^* = I$ and the matrix is unitary. □

The proves follow the idea as described in [DSa 96]. Another way to prove Lemma 2.7 is given in [BV 97].

Doubly Stochastic Matrixes

Doubly stochastic matrixes stand for transition matrixes for probabilistic reversible Automata considered in this thesis.

Definition 2.8. *A real $(n \times n)$ matrix S, $s_{i,j} \geq 0$, is called* stochastic*, if* $\forall j \; \sum_{i=1}^{n} s_{i,j} = 1$.

Definition 2.9. *A stochastic $n \times n$ matrix D is called* doubly stochastic*, if* $\forall i \; \sum_{j=1}^{n} d_{i,j} = 1$.

Lemma 2.10. *If matrixes A and B are doubly stochastic, then their direct product is a doubly stochastic matrix.*

Lemma 2.11. *If A is a doubly stochastic matrix and X - a vector with components $x_i \geq 0$, then* $\max(X) \geq \max(AX)$ *and* $\min(X) \leq \min(AX)$.

Proof. Let us consider $X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$ and $A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$, where A is doubly stochastic. Let us suppose that $x_j = \max(X)$. For any i, $1 \leq i \leq n$,

$$x_j = a_{i1}x_j + a_{i2}x_j + \ldots + a_{in}x_j \geq a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n.$$

Therefore x_j is greater or equal than any component of AX. The second inequality is proved in the same way. $\square$

Definition 2.12. *We say that a doubly stochastic matrix S is* unitary stochastic ([MO 79])*, if exists a unitary matrix U such that $\forall i, j$ $|u_{i,j}|^2 = s_{i,j}$.*

Remark 2.13. Not every doubly stochastic matrix is unitary stochastic. Such matrix is, for example, $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$.

2.2 Markov Chains

We recall several definitions from the theory of finite Markov chains ([KS 76], etc.) used in this thesis when describing behavior of PRA.
A Markov chain with n states can be determined by an $n \times n$ stochastic matrix A, i.e., matrix, where the sum of elements of every column in the matrix is 1. If $A_{i,j} = p > 0$, it means that a state q_i is accessible from a state q_j with a positive probability p in one step.

2.2.1 Classification of states

Definition 2.14. *A state q_j is accessible from q_i (denoted $q_i \to q_j$) if there is a positive probability to get from q_i to q_j in 1 or more steps.*

Note that some authors consider zero steps are valid for this definition that means $q_i \to q_i$ for any i, we do not.

Definition 2.15. *States q_i and q_j communicate (denoted $q_i \leftrightarrow q_j$) if $q_i \to q_j$ and $q_j \to q_i$.*

For accessibility or communication in one step we will put the corresponding matrix above the symbol. Example: $q_i \xrightarrow{A} q_j$ means there is a positive probability to get from q_i to q_j in 1 step. Or the same $A_{j,i} > 0$. We will use also word above the arrow instead of matrix in this notation in the thesis.

Definition 2.16. *A state q is called recurrent if $\forall i \ q \rightarrow q_i \Rightarrow q_i \rightarrow q$. Otherwise the state is called transient.*

There several different definitions for transient states proven to be equivalent to the above, important for us is

Definition 2.17. *A state q_i is called transient iff $\sum\limits_{n\rightarrow\infty} (A^n)_{i,i} < \infty$*

Definition 2.18. *A state q is called absorbing if there is a zero probability of exiting from this state.*

Definition 2.19. *A Markov chain without transient states is called irreducible if for all q_i, q_j $q_i \leftrightarrow q_j$. Otherwise the chain without transient states is called reducible.*

Definition 2.20. *The period of an recurrent state $q_i \in Q$ of a Markov chain with a matrix A is defined as $d(q_i) = \gcd\{n > 0 \mid (A^n)_{i,i} > 0\}$.*

Definition 2.21. *An recurrent state q_i is called aperiodic if $d(q_i) = 1$. Otherwise the recurrent state is called periodic.*

Definition 2.22. *A Markov chain without transient states is called aperiodic if all its states are aperiodic. Otherwise the chain without transient states is called periodic.*

Definition 2.23. *Markov chain is called absorbing iff that contains at least one absorbing state, and for any non-absorbing state q_i there is an absorbing state that is accessible from q_i. Thus the states of absorbing Markov Chain can be numbered so that transition matrix A has a form*

$$\begin{matrix} A & O \\ B & I \end{matrix}.$$

where I - unit matrix, O - all zero matrix

Definition 2.24. *A probability distribution X of a Markov chain with a matrix A is called stationary, if $AX = X$.*

2.2.2 Behavior of Markov chains

We recall the following theorem from the theory of finite Markov chains about stationary distribution:

Theorem 2.25. *If a Markov chain with a matrix A is irreducible and aperiodic, then*
a) it has a unique stationary distribution Z;
b) $\lim\limits_{n\to\infty} A^n = (Z, \dots, Z)$;
c) $\forall X \lim\limits_{n\to\infty} A^n X = Z$.

We recall the following fact regarding transient states of a Markov Chain

Theorem 2.26. *Given Markov chain with matrix A and transient state q_i, for matrix A^n when $n \to \infty$, $a^n_{ij} \to 0$ for any j.*

Proof. Follows from Definition 2.17 □

2.2.3 Doubly Stochastic Markov Chains

The notion is introduced according to the needs of PRA and used in proofs on forbidden constructions.

Definition 2.27. *A Markov chain is called doubly stochastic, if its transition matrix is a doubly stochastic matrix.*

Corollary 2.28. *If a doubly stochastic Markov chain with an $m \times m$ matrix A is irreducible and aperiodic,*
a) $\lim\limits_{n\to\infty} A^n = \begin{pmatrix} \frac{1}{m} & \dots & \frac{1}{m} \\ \dots & \dots & \dots \\ \frac{1}{m} & \dots & \frac{1}{m} \end{pmatrix}$;
b) $\forall X \lim\limits_{n\to\infty} A^n X = \begin{pmatrix} \frac{1}{m} \\ \dots \\ \frac{1}{m} \end{pmatrix}$.

Proof. By Theorem 2.25. □

Lemma 2.29. *If M is a doubly stochastic Markov chain with a matrix A, then $\forall q\ q \to q$.*

Proof. Assume existence of q_0 such that q_0 is not accesible from itself. Let $Q_{q_0} = \{q_i \mid q_0 \to q_i\} = \{q_1, \dots, q_k\}$. Q_{q_0} is not empty set. Consider those rows and columns of A, which are indexed by states in Q_{q_0}. These rows and columns form a submatrix A'. Each column j of A' must include all non-zero elements of the corresponding column of A as those states are accesible from the state q_j, hence also from q_0 and are in Q_{q_0}. Therefore $\forall j,\ 1 \leq j \leq k$, $\sum_{i=1}^{k} A'_{i,j} = 1$ and $\sum_{1\leq i,j\leq k} A'_{i,j} = k$. On the other hand, since $q_0 \notin Q_{q_0}$, a row of A' indexed by a state accesible in one step from q_0 does not include all nonzero elements. Since A is doubly stochastic, $\exists i, 1 \leq i \leq k,\ \sum_{j=1}^{k} A'_{i,j} < 1$ and $\sum_{1\leq i,j\leq k} A'_{i,j} < k$. This is a contradiction. ☐

Corollary 2.30. *Suppose A is a doubly stochastic matrix. Then exists $k > 0$, such that $\forall i\ (A^k)_{i,i} > 0$.*

Proof. Consider an $m \times m$ doubly stochastic matrix A. By Lemma 2.29, $\forall i$ $\exists n_i > 0\ (A^{n_i})_{i,i} > 0$. Take $n = \prod_{s=1}^{m} n_s$. For every i, $(A^n)_{i,i} > 0$. ☐

Lemma 2.31. *If M is a doubly stochastic Markov chain with a matrix A, then $\forall q_a, q_b\ A_{b,a} > 0 \Rightarrow q_b \to q_a$.*

Proof. $A_{b,a} > 0$ means that q_b is accesible from q_a in one step. We have to prove, that $q_b \to q_a$. Assume from the contrary, that q_a is not accesible from q_b. Let $Q_{q_b} = \{q_i \mid q_b \to q_i\} = \{q_1, q_2, \ldots, q_k\}$. By Lemma 2.29, $q_b \in Q_{q_b}$. As in proof of Lemma 2.29, consider a matrix A', which is a submatrix of A and whose rows and columns are indexed by states in Q_{q_b}. Each column j has to include all nonzero elements of the corresponding column of A. Therefore $\forall j,\ 1 \leq j \leq k,\ \sum_{i=1}^{k} A'_{i,j} = 1$ and $\sum_{1\leq i,j\leq k} A'_{i,j} = k$. On the other hand, $A_{b,a} > 0$ and $q_a \notin Q_{q_b}$, therefore a row of A' indexed by q_b does not include all nonzero elements. Since A is doubly stochastic, $\sum_{j=1}^{k} A'_{b,j} < 1$ and $\sum_{1\leq i,j\leq k} A'_{i,j} < k$. This is a contradiction. ☐

Corollary 2.32. *If M is a doubly stochastic Markov chain and $q_a \to q_b$, then $q_a \leftrightarrow q_b$.*

Proof. If $q_a \to q_b$ then exists a sequence $q_{i_1}, q_{i_2}, \ldots, q_{i_k}$, such that $A_{i_1,a} > 0,\ A_{i_2,i_1} > 0, \ldots, A_{i_k,i_{k-1}} > 0,\ A_{b,i_k} > 0$. By Lemma 2.31, we get $q_b \to q_{i_k}$, $q_{i_k} \to q_{i_{k-1}}$, ..., $q_{i_2} \to q_{i_1}$, $q_{i_1} \to q_a$. Therefore $q_b \to q_a$. ☐

Lemma 2.33. *Suppose A is a doubly stochastic matrix and $k > 0$, such that $\forall i\ (A^k)_{i,i} > 0$. Then exist $m > 0$ such that for all pairs $q_i\ q_j$ where $q_i \to q_j$ for A^k, $q_i \to q_j$ in one step for (A^{k*m})*

Proof. Assume $q_i \to q_j$ in x steps, as according to Lemma 2.30 $q_i \to q_i$ in one step, $q_i \to q_j$ in x+1 step as well. For any pair of states q_i q_j where $q_i \to q_j$ for A^k, q_j is accessible in less then n number of rows in A steps. Thus $m = n$ gives the necessary constant. □

Corollary 2.34. *Accessibility is a class property for states of doubly stochastic Markov chains.*

Proof.

- reflexive - $\forall i q_i \to q_i$ by Lemma 2.29;
- symmetric - If $q_i \to q_j$ then $q_j \to q_i$ by Corollary 2.32;
- transitive - if $q_i \to q_j$ and $q_j \to q_k$ then $q_i \to q_k$.

□

Corollary 2.35. *Communication is a class property for states of doubly stochastic Markov chains.* [1]

Therefore, for doubly stochastic Markov chains both communication and accessibility divides the state space into mutually disjoint exclusive classes.

2.3 Automata

In this section, we define notions applicable to arbitrary type of automata we will use through out the thesis. That basically follows the description of automata given in [Go 02].

Abstract Automaton

Consider an abstract automaton $A = (Q, \Sigma_1, \ldots, \Sigma_m, q_0, \delta)$, where Q is a finite set of states, Σ_k is an alphabet of the k-th tape, q_0 is the initial state and δ is a transition function. (See Figure 2.1.)

Each tape is potentially infinite on both directions. The cells of each tape are indexed by numbers in $\mathbb{Z}$. Each cell of the k-th tape stores a symbol in Σ_k or white space, denoted λ. A cell the k-th tape head is above is called *the k-th current cell.* The transition function determines possible transitions of the automaton depending on its current configuration.

Definition 2.36. *A* configuration *of an abstract automaton is* $c = (q_i, n_1, \sigma_1, \tau_1, \ldots, n_m, \sigma_m, \tau_m)$*, where the automaton is in a state* $q_i \in Q$

[1] if we would use definition of accessibility where a state is always accessible from itself then this corollary would hold for any finite Markov chain

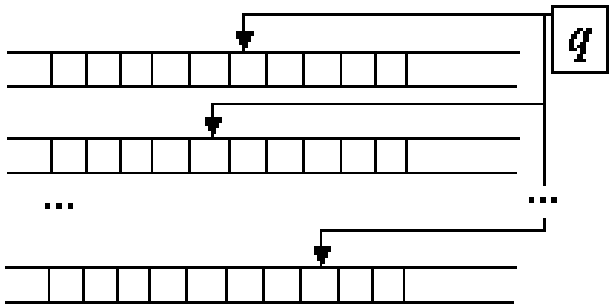

Figure 2.1: An abstract automaton

and $\sigma_k \tau_k \in \Sigma_k^*$ *is a finite word on the k-th input tape. The k-th current cell is indexed by* n_k *and it contains the last symbol of the word* σ_k*, if* $\sigma_k \neq \epsilon$ *and* λ*, otherwise. All cells before or after* $\sigma_k \tau_k$ *are blank (contain* λ*).*

The automaton operates in discreet time moments $(t_0, \ldots, t_r, \ldots)$. If the automaton cannot change contents of a particular tape, it is called *input tape*. Let us assume that the automaton has p input tapes, and renumber the tapes, so that first come input tapes. At the time moment t_0, the automaton is in configuration $(q_0, 0, \epsilon, \tau_1, \ldots, 0, \epsilon, \tau_p, 0, \epsilon, \epsilon, \ldots, 0, \epsilon, \epsilon)$, where $\tau_1, \ldots, \tau_p$ are *input words*. We refer to the input word tuple as *input*. At each time moment, the automaton performs a single transition, called *step*. At each step, depending on its current state and symbols in current cells, the automaton may change its current state, change the contents of current cells, and afterwards, move each tape head one cell forward or backward.

Formally, the transition function δ defines a binary relation ρ from the set $Q \times \Sigma_1 \times \ldots \times \Sigma_m$ to the set $Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m$. $(q_1, s_1, \ldots, s_m)\rho(q_2, s'_{p+1}, \ldots, s'_m, d_1, \ldots, d_m)$, $d_i \in \{\leftarrow, \downarrow, \rightarrow\}$, means that for the automaton being in the state q_1 and having symbols $s_1, \ldots, s_m$ in current cells, the following transition is possible: the automaton goes to the state q_2, writes $s'_{p+1}, \ldots, s'_m$ into the current cells of the tapes $p+1, \ldots, m$ and moves tape heads according to the directions d_i. If this relation is a function, we speak about *deterministic automata*, other considered possibilities are *probabilistic automata* and *quantum automata*. Probabilistic automata perform transitions with certain probabilities, whereas quantum automata - with certain *amplitudes*.

For technical reasons, we may introduce two categories of white spaces for input tapes, called *end-markers*; one is used before input word and denoted as $\looparrowright$, and the other after input word and denoted as $\looparrowleft$. So every input word is enclosed into end-marker symbols $\looparrowright$ and $\looparrowleft$[2]. Therefore we introduce a

[2]To get rid of infinite input tapes we may also assume that input tapes are circular

working alphabet of the k-th input tape as $\Gamma_k = \Sigma_k \cup \{\looparrowright, \looparrowleft\}$. We define the *length of input* as the length of the longest word in the input word tuple (including one end-marker to the left of the word and one to the right of the word).

By $\mathbf{C}$ we denote the set of all configurations of an automaton. This set is countably infinite.

Remark 2.37. It is possible to reach only a finite number of other configurations from a given configuration in one step, all the same, within one step the given configuration is reachable only from a finite number of different configurations.

An abstract automaton introduced above is actually a description of an m-tape Turing machine. To define other types of automata, we apply specific restrictions to this general model. We say that an automaton is *1-way*, if at each step, it must move each input tape head one cell forward. We say that an automaton is *1.5-way*, if at each step, it may not move input tape heads backward. Otherwise, an automaton is called *2-way*. We refer to an automaton as a *finite automaton*, if all of its tapes are input tapes.

To halt computation of the automaton, we may consider at least two options. According to the first option, a subset of $\mathbf{C}$ is introduced and configurations in the subset are marked as *halting* configurations. We monitor the computation of the automaton and stop the computation as soon as the automaton enters a halting configuration. According to the second option, we determine the number of steps of computation in advance, and run the automaton the specified number of steps. In particular, when the number of steps is equal to the length of input, we get *real-time* automata.

Word Acceptance

We study automata in terms of formal languages they recognize. At least two definitions exist, how to interpret word acceptance, and hence, language recognition, for automata.

Definition 2.38. "Decide and halt" acceptance. *Consider an automaton with the set of configurations partitioned into non-halting configurations and halting configurations, where halting configurations are further classified as*

---

and the length of every input tape is $l = \max\limits_{0<k\leq p} \{|\tau_k|\} + 2$, so that the next cell after the cell indexed by $l-1$ is the cell indexed by 0. The cells indexed by 0 store $\looparrowright$ and the rest blank cells store $\looparrowleft$.

accepting configurations and rejecting configurations. We say that an automaton accepts (rejects) an input in a decide-and-halt manner, if the following conditions hold:

- *the computation is halted as soon as the automaton enters a halting configuration;*
- *if the automaton enters an accepting configuration, the input is accepted;*
- *if the automaton enters a rejecting configuration, the input is rejected.*

We refer to the decide-and-halt automata as DH-automata further in the thesis. In case of real-time automata, we may use the following definition.

Definition 2.39. Classical acceptance. *Consider an automaton with the set of configurations partitioned into accepting configurations and rejecting configurations. We say that an automaton accepts (rejects) an input classically, if the following conditions hold:*

- *the computation is halted as soon as the number of computation steps is equal to the length of input;*
- *if the automaton has entered an accepting configuration when halted, the input is accepted;*
- *if the automaton has entered a rejecting configuration when halted, the input is rejected.*

We refer to the classical acceptance automata as classical automata or C-automata further in the thesis.

The both definitions generally are not equivalent.

Language Recognition

Having defined word acceptance, we define language recognition in an equivalent way as in [R 63].

By $p_{x,A}$ we denote the probability that an input x is accepted by an automaton A.

Furthermore, we denote $P_L = \{p_{x,A} \mid x \in L\}$, $\overline{P_L} = \{p_{x,A} \mid x \notin L\}$, $p_1 = \sup \overline{P_L}$, $p_2 = \inf P_L$.

Definition 2.40. *We say that an automaton A recognizes a language L with interval (p_1, p_2), if $p_1 \leq p_2$ and $P_L \cap \overline{P_L} = \emptyset$.*

Definition 2.41. *We say that an automaton A recognizes a language L with bounded error and interval (p_1, p_2), if $p_1 < p_2$.*

We consider only bounded error language recognition in this thesis.

Definition 2.42. *An automaton recognizes a language with probability p if the automaton recognizes the language with interval $(1-p, p)$.*

Definition 2.43. *We say that a language is recognized by some class of automata with probability $1-\varepsilon$, if for every $\varepsilon > 0$ there exists an automaton in the class which recognizes the language with interval $(\varepsilon_1, 1-\varepsilon_2)$, where $\varepsilon_1, \varepsilon_2 \leq \varepsilon$.*

Quantum Automata

In case of a quantum automaton, the transition function is

$$\delta : (Q \times \Sigma_1 \times \ldots \times \Sigma_m) \times (Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m) \longrightarrow \mathbb{C}_{[0,1]}.$$

On each computation step in case of pure state automaton, the quantum automaton is in quantum superposition of configurations. In case of mixed state automaton, the automaton with certain probabilities is in one of several possible quantum superpositions, or in a mixed state. $|\psi\rangle = \sum_{c \in \mathbf{C}} \alpha_c |c\rangle$, where $\sum_{c \in \mathbf{C}} |\alpha_c|^2 = 1$ and $\alpha_c \in \mathbb{C}$ is the amplitude of a configuration $|c\rangle$. Every configuration $|c\rangle \in \mathbf{C}$ is a basis vector in the Hilbert space H, determined by $l_2(\mathbf{C})$. Every quantum automaton defines a linear operator (evolution) over this Hilbert space. Due to the laws of quantum mechanics, this operator must be unitary. Although evolution operator matrix is infinite, by Remark 2.37 it has a finite number of nonzero elements in each row and column, therefore it is possible to derive necessary and sufficient conditions, i.e., *well-formedness conditions* to check unitarity for each particular automata type.

General measurements. After each step, a *measurement* is applied to the current quantum superposition of configurations. A measurement is defined as follows. We introduce a set partition of $\mathbf{C}$ as $\{\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C_z}\}$. So $\bigcup_{0<i\leq z} \mathbf{C_i} = \mathbf{C}$ and if $i \neq j$ then $\mathbf{C_i} \cap \mathbf{C_j} = \emptyset$. $E_1, E_2, \ldots, E_z$ are subspaces of H spanned by $\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C_z}$, respectively. We use the observable $\mathcal{O}_1$ that corresponds to the orthogonal decomposition $H = E_1 \oplus E_2 \oplus \ldots \oplus E_z$. If the quantum superposition before the observation is $\sum_{c \in \mathbf{C}} \alpha_c |c\rangle$, with probability $p_i = \sum_{c \in \mathbf{C_i}} |\alpha_c|^2$ the outcome of the observation is $|\psi_i\rangle = \frac{1}{\sqrt{p_i}} \sum_{c \in \mathbf{C_i}} \alpha_c |c\rangle$. Hence the total outcome of the observation is a *mixed state* $\sum_{i=1}^{z} p_i |\psi_i\rangle\langle\psi_i|$.

Thus if $z = 1$, we get quantum automaton with pure states as all the time automaton is only in single superposition, otherwise we generally have quantum automaton with mixed states. We get other marginal case, when $\mathbf{C}$ is set partitioned into infinitely many subsets, with a single configuration in each subset[3]. In that case, the resulting quantum automaton is a special kind of a probabilistic automaton. See the next subsection for further details.

Word acceptance measurements. Another type of measurement is applied to the quantum automaton to facilitate language recognition.

Decide-and-halt acceptance. We have to monitor when the quantum automaton enters a halting configuration. Hence we perform the following measurement after each step. We partition $\mathbf{C}$ as $\mathbf{C_a}$, $\mathbf{C_r}$ and $\mathbf{C_{non}}$, i.e., accepting, rejecting and non-halting configurations. E_a, E_r and E_{non} are subspaces of H spanned by $\mathbf{C_a}$, $\mathbf{C_r}$, and $\mathbf{C_{non}}$, respectively. We use the observable $\mathcal{O}_2$ that corresponds to the orthogonal decomposition $H = E_a \oplus E_r \oplus E_{non}$. The outcome of each observation is either "accept" or "reject" or "continue". If the quantum superposition before the observation is $\sum_{c \in \mathbf{C}} \alpha_c |c\rangle$, with probability $p_a = \sum_{c \in \mathbf{C_a}} |\alpha_c|^2$ the input is accepted, with probability $p_r = \sum_{c \in \mathbf{C_r}} |\alpha_c|^2$ the input is rejected, and with probability $p_{non} = \sum_{c \in \mathbf{C_{non}}} |\alpha_c|^2$ the automaton is in the quantum superposition of non-halting states $|\psi\rangle = \frac{1}{\sqrt{p_{non}}} \sum_{c \in \mathbf{C_{non}}} \alpha_c |c\rangle$.

Classical acceptance. After the computation is halted, we have to determine, whether the automaton has entered accepting or rejecting configuration. We partition $\mathbf{C}$ as $\mathbf{C_{acc}}$ and $\mathbf{C_{rej}}$, i.e., accepting and rejecting configurations. E_{acc}, E_{rej} are subspaces of H spanned by $\mathbf{C_{acc}}$ and $\mathbf{C_{rej}}$, respectively. We use the observable $\mathcal{O}_3$ that corresponds to the orthogonal decomposition $H = E_{acc} \oplus E_{rej}$. The outcome of the observation is either "accept" or "reject". If the quantum superposition before the observation is $\sum_{c \in \mathbf{C}} \alpha_c |c\rangle$, with probability $p_{acc} = \sum_{c \in \mathbf{C_{acc}}} |\alpha_c|^2$ the input is accepted and with probability $p_{rej} = \sum_{c \in \mathbf{C_{rej}}} |\alpha_c|^2$ the input is rejected.

In case both general measurement and word acceptance measurement have to be performed in a single step, it is easy to see that the order of measurements is irrelevant, actually both measurements may be combined into a single measurement after each step.

Putting things together, each computation step consists of two parts. At first the unitary evolution operator is applied to the current quantum

[3]By Remark 2.37, on each computation step the number of configurations in a quantum superposition is finite, so on each step it is possible to make the corresponding measurement actually using some finite partition of $\mathbf{C}$.

superposition and then the appropriate measurements are applied, using observables as defined above.

Probabilistic Automata

In case of probabilistic automaton the transition function is

$$\delta : (Q \times \Sigma_1 \times \ldots \times \Sigma_m) \times (Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m) \longrightarrow \mathbb{R}_{[0,1]}.$$

After its every step, the probabilistic automaton is in some probability distribution $p_0c_0 + p_1c_1 + \ldots + p_zc_z$, where $p_0 + p_1 + \ldots + p_z = 1$. Such probability distribution to be called a superposition of configurations as analogy to the quantum case.

A linear closure of **C** forms a linear space, where every configuration can be viewed as a basis vector. This basis is called a *canonical basis*. Every probabilistic automaton defines a linear operator (evolution) over this linear space. The linear operator is defined by the probability transition matrixes.

Deterministic Reversible Automata

Deterministic reversible automata can be viewed as a special case of quantum automata. The transition function is

$$\delta : (Q \times \Sigma_1 \times \ldots \times \Sigma_m) \times (Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m) \longrightarrow \{0, 1\}.$$

The condition of unitarity in the deterministic case implies that for any configuration there is exactly one configuration from which we get there in one step. So in the operator of automata there is exactly one 1 in every row and every column.

Automata Notations

We regard quantum automata, probabilistic reversible automata and deterministic reversible automata as reversible automata. Refering to different types of automata, we shall use the following notation:

$$\text{[C|DH-]}\langle\text{automata type}\rangle\text{[-P|M]}.$$

C refers to "classical", whereas DH refers to "decide-and-halt". Notations P and M are used in the case of quantum automata. P denotes an automaton with pure states, whereas M - an automaton with mixed states.

For example, C-QFA-M are quantum finite automata with mixed states, using classical definition of language recognition.

Abbreviations for different automata types are: QFA - quantum finite automaton, Q1CA - quantum 1 counter automaton, PRA - probabilistic reversible automaton, DRA - deterministic reversible automaton.

Chapter 3

Probabilistic reversible automata

3.1 Probabilistic Reversible Automata

Let us consider A.Nayak's model of quantum automata with mixed states, [N 99]. A variety of this model for arbitrary type of automata was considered in the previous subsection. (The difference is that Nayak's model allows a fixed sequence of unitary transformations and subsequent measurements after each step.) As noted there, if a result of every observation is a single configuration, not a superposition of configurations, we actually get a probabilistic automaton. However, the following property applies to such probabilistic automata - their evolution matrixes are *doubly* stochastic.

So we give the following definition for probabilistic reversible automata:

Definition 3.1. *A probabilistic automaton is called* reversible *if its evolution is described by a doubly stochastic matrix, using canonical basis.*

If the evolution of a probabilistic reversible automaton is described by *unitary stochastic* matrix (see Definition 2.12), the automaton can be viewed as a special case of a quantum automaton with mixed states.

It is necessary to note that in [AF 98], A. Ambainis and R. Freivalds proposed a more restricted notion of probabilistic reversibility, that allowed probabilistic choice only at the first step and after that automaton acts as deterministic reversible automaton. For example, they show that for the language $L = \{a^{2n+3} | n \in \mathbb{N}\}$, not recognizable by a 1-way deterministic reversible finite automata, there exists a 1-way probabilistic reversible finite automaton which recognizes the language. Consequently, this restricted notion was used in [YKTI 00]. That model is actually a special case of probabilistic reversible DH-automata, as defined in the thesis.

3.2 Definition of 1-way Probabilistic Reversible Automata

In this section we give formal definition of 1-way PRA.

Definition 3.2. *1-way probabilistic reversible automaton (PRA)* $A = (Q, \Sigma, q_0, \delta)$ *is specified by a finite set of states* Q*, a finite input alphabet* Σ*, an initial state* $q_0 \in Q$*, and a transition function*

$$\delta : Q \times \Gamma \times Q \longrightarrow \mathbb{R}_{[0,1]},$$

where $\Gamma = \Sigma \cup \{\looparrowright, \looparrowleft\}$ *is the input tape alphabet of* A *and* $\looparrowright$, $\looparrowleft$ *are end-markers not in* Σ*. Furthermore, transition function satisfies the following requirements:*

$$\forall (q_1, \sigma_1) \in Q \times \Gamma \ \sum_{q \in Q} \delta(q_1, \sigma_1, q) = 1 \tag{3.1}$$

$$\forall (q_1, \sigma_1) \in Q \times \Gamma \ \sum_{q \in Q} \delta(q, \sigma_1, q_1) = 1 \tag{3.2}$$

For every input symbol $\sigma \in \Gamma$, the transition function may be determined by a $|Q| \times |Q|$ matrix V_σ, where $(V_\sigma)_{i,j} = \delta(q_j, \sigma, q_i)$.

Lemma 3.3. *All matrixes* V_σ *are doubly stochastic iff conditions (3.1) and (3.2) of Definition 3.2 hold.*

Proof. Trivial. □

A linear operator U_A corresponds to the automaton A. Formal definition of this operator follows:

Definition 3.4. *Given a configuration* $c = \langle \nu_i q_j \sigma \nu_k \rangle$,

$$U_A c \stackrel{\text{def}}{=} \sum_{q \in Q} \delta(q_j, \sigma, q) \langle \nu_i \sigma q \nu_k \rangle.$$

Given a superposition of configurations $\psi = \sum_{c \in C} p_c c$,

$$U_A \psi \stackrel{\text{def}}{=} \sum_{c \in C} p_c U_A c.$$

Using canonical basis, U_A is described by an infinite matrix M_A.
To comply with Definition 3.1, we have to state the following:

Lemma 3.5. *Matrix M_A is doubly stochastic iff conditions (3.1) and (3.2) of Definition 3.2 hold.*

Proof. Condition (3.1) takes place if and only if the sum of elements in every column in M_A equal to 1. Condition (3.2) takes place if and only if the sum of elements in every row in M_A equal to 1. □

This completes our formal definition of 1 way PRA.

3.3 1-way Probabilistic Reversible C Automata

3.3.1 Definition

For a definition 3.1 we define word acceptance as specified in Definition 2.39. The set of accepting states is Q_F and the set of rejecting states is $Q \setminus Q_F$. We define language recognition as in Definition 2.42. That completes formal definition of 1-way C-PRA automata.

3.3.2 Boost of Probability and Closure properties

Lemma 3.6. *If a language is recognized by a C-PRA A with interval (p_1, p_2), exists a C-PRA which recognizes the language with probability p, where*

$$p = \begin{cases} \frac{p_2}{p_1+p_2}, & \text{if } p_1 + p_2 \geq 1 \\ \frac{1-p_1}{2-p_1-p_2}, & \text{if } p_1 + p_2 < 1. \end{cases}$$

Proof. Let us assume, that the automaton A has $n-1$ states. We shall consider the case $p_1 + p_2 > 1$.

Informally, having read endmarker symbol $\hookleftarrow$, we simulate the automaton A with probability $\frac{1}{p_1+p_2}$ and reject input with probability $\frac{p_1+p_2-1}{p_1+p_2}$.

Formally, to recognize the language with probability $\frac{p_2}{p_1+p_2}$, we modify the automaton A. We add a new state $q_r \notin Q_F$, and change the transition function in the following way:

- $\forall\sigma$, $\sigma \neq \hookleftarrow$, $\delta(q_r, \sigma, q_r) \stackrel{\text{def}}{=} 1$;
- $\delta(q_0, \hookleftarrow, q_r) \stackrel{\text{def}}{=} \frac{p_1+p_2-1}{p_1+p_2}$;
- $\forall q$, $q \neq q_r$, $\delta(q_0, \hookleftarrow, q) \stackrel{\text{def}}{=} \frac{1}{p_1+p_2}\delta_{old}(q_0, \hookleftarrow, q)$.

Now the automaton has n states. Since endmarker symbol $\looparrowright$ is read only once at the beginning of an input word, we can disregard the rest of transition function values, associated with $\looparrowright$: $\forall q_i, q_j$, where $q_i \neq q_0$, $\delta(q_i, \looparrowright, q_j) \stackrel{\text{def}}{=} \frac{1-\delta(q_0,\looparrowright,q_j)}{n-1}$.

The transition function satisfies the requirements of Definition 3.2 and the constructed automaton recognizes the language with probability $\frac{p_2}{p_1+p_2}$.

The case $p_1 + p_2 < 1$ is very similar. Informally, having read endmarker symbol $\looparrowright$, we simulate the automaton A with probability $\frac{1}{2-p_1-p_2}$ and accept input with probability $\frac{1-p_1-p_2}{2-p_1-p_2}$. □

Theorem 3.7. *If a language is recognized by a C-PRA, it is recognized by C-PRA with probability* $1 - \varepsilon$.

Proof. Following Lemma 3.6, we can assume that a language L is recognized by a C-PRA automaton $A = (Q, \Sigma, q_0, Q_F, \delta)$ with probability p.

Let us consider a system of m copies of the automaton A, denoted as A_m. We shall say that our system has accepted (rejected) a word if more (less or equal) than a half of automata in the system have accepted (rejected) the word. We define language recognition as in Definition 2.41.

Let us consider a word $\omega \in L$. The automaton A accepts ω with probability $p_\omega \geq p$. As a result of reading ω, μ_m^ω automata of the system will accept the word, and the rest will reject it. The system has accepted the word, if $\frac{\mu_m^\omega}{m} > \frac{1}{2}$. Let us take η_0, such that $0 < \eta_0 < p - \frac{1}{2} \leq p_w - \frac{1}{2}$. Estimating the probability that $\frac{\mu_m^\omega}{m} > \frac{1}{2}$, we have

$$P\left\{\frac{\mu_m^\omega}{m} > \frac{1}{2}\right\} \geq P\left\{p_\omega - \eta_0 < \frac{\mu_m^\omega}{m} < p_\omega + \eta_0\right\} = P\left\{\left|\frac{\mu_m^\omega}{m} - p_\omega\right| < \eta_0\right\} \tag{3.3}$$

In case of m Bernoulli trials, Chebyshev's inequality may be used to prove the following ([GS 97], p. 312):

$$P\left\{\left|\frac{\mu_m^\omega}{m} - p_\omega\right| \geq \eta_0\right\} \leq \frac{p_\omega(1-p_\omega)}{m\eta_0^2} < \frac{1}{4m\eta_0^2} \tag{3.4}$$

The last inequality induces that

$$P\left\{\left|\frac{\mu_m^\omega}{m} - p_\omega\right| < \eta_0\right\} > 1 - \frac{1}{4m\eta_0^2} \tag{3.5}$$

Finally, putting (3.3) and (3.5) together,

$$P\left\{\frac{\mu_m^\omega}{m} > \frac{1}{2}\right\} > 1 - \frac{1}{4m\eta_0^2} \tag{3.6}$$

Inequality (3.6) is true for every $\omega \in L$.

On the other hand, let us consider a word $\xi \notin L$. The automaton A accepts ξ with probability $p_\xi \leq 1 - p$. If we take the same η_0, $0 < \eta_0 < p - \frac{1}{2} \leq \frac{1}{2} - p_\xi$ and for every ξ we have

$$P\left\{\frac{\mu_m^\xi}{m} > \frac{1}{2}\right\} \leq P\left\{\left|\frac{\mu_m^\xi}{m} - p_\xi\right| \geq \eta_0\right\} < \frac{1}{4m\eta_0^2} \tag{3.7}$$

Due to (3.6) and (3.7), for every $\varepsilon > 0$, if we take $n > \frac{1}{4\varepsilon\eta_0^2}$, we get a system A_n which recognizes the language L with interval $(\varepsilon_1, 1 - \varepsilon_2)$, where $\varepsilon_1, \varepsilon_2 < \varepsilon$.

Let us show that A_n can be simulated by a C-PRA. The automaton $A' = (Q', \Sigma, q_0', Q_F', \delta')$ is constructed as follows:

$Q' \stackrel{\text{def}}{=} \{\langle q_{s_1} q_{s_2} \dots q_{s_n}\rangle \mid 0 \leq s_i \leq |Q| - 1\}$; $q_0' \stackrel{\text{def}}{=} \langle q_0 q_0 \dots q_0\rangle$.

A sequence $\langle q_{s_1} q_{s_2} \dots q_{s_n}\rangle$ is an accepting state of A' if more than a half of elements of the sequence are accepting states of A. We have defined the set Q_F'.

Given $\sigma \in \Gamma$, $\delta'(\langle q_{a_1} q_{a_2} \dots q_{a_n}\rangle, \sigma, \langle q_{b_1} q_{b_2} \dots q_{b_n}\rangle) \stackrel{\text{def}}{=} \prod_{i=1}^{n} \delta(q_{a_i}, \sigma, q_{b_i})$.

In essence, Q' is n-th Cartesian power of Q and the linear space formed by A' is n-th tensor power of the linear space formed by A. If we take a symbol $\sigma \in \Gamma$, transition is determined by $|Q|^n \times |Q|^n$ matrix V_σ', which is n-th matrix direct power of V_σ, i.e, $V_\sigma' = \bigotimes_{i=1}^{n} V_\sigma$.

A' simulates the system A_n. Since matrix direct product of two doubly stochastic matrixes is a doubly stochastic matrix, $\forall\sigma$ V_σ' are doubly stochastic matrixes. Therefore our automaton A' is C-PRA.

We have proved that $\forall\varepsilon > 0$ the language L is recognized by some C-PRA with interval $(\varepsilon_1, 1 - \varepsilon_2)$, where $\varepsilon_1, \varepsilon_2 < \varepsilon$. Therefore, by Lemma 3.6, the language L is recognized with probability $1 - \varepsilon$. □

Lemma 3.8. *If a language L_1 is recognizable with probability greater than $\frac{2}{3}$ and a language L_2 is recognizable with probability greater than $\frac{2}{3}$ then languages $L_1 \cap L_2$ and $L_1 \cup L_2$ are recognizable with probability greater than $\frac{1}{2}$.*

Proof. Let us consider automata $A = (Q_A, \Sigma, q_{0,A}, Q_{F,A}, \delta_A)$ and $B = (Q_B, \Sigma, q_{0,B}, Q_{F,B}, \delta_B)$ which recognize the languages L_1, L_2 with probabilities $p_1, p_2 > \frac{2}{3}$, respectively. Let us assume that A, B have m and n states, respectively. Without loss of generality we can assume that $p_1 \leq p_2$.

Informally, having read endmarker symbol $\looparrowright$, with probability $\frac{1}{2}$ we simulate the automaton A_1 and with the same probability we simulate the automaton A_2.

Formally, we construct an automaton $C = (Q, \Sigma, q_0, Q_F, \delta)$ with the following properties.
$Q \stackrel{\text{def}}{=} Q_A \cup Q_B$; $q_0 \stackrel{\text{def}}{=} q_{0,A}$; $Q_F \stackrel{\text{def}}{=} Q_{F,A} \cup Q_{F,B}$; $\delta \stackrel{\text{def}}{=} \delta_A \cup \delta_B$, with an exception that:

- $\delta(q_0, \looparrowright, q_{i,A}) = \frac{1}{2}\delta_A(q_0, \looparrowright, q_{i,A})$;
- $\delta(q_0, \looparrowright, q_{i,B}) = \frac{1}{2}\delta_B(q_0, \looparrowright, q_{i,B})$;
- $\forall q_i,\ q_i \neq q_0,\ \delta(q_i, \looparrowright, q_j) = \frac{2-\delta(q_0,\looparrowright,q_j)}{m+n-1}$.

Since δ satisfies Definition 3.2, our construction of C-PRA is complete.

The automaton C recognizes the language $L_1 \cap L_2$ with interval $(p, \frac{p_1+p_2}{2})$, where $p \leq 1 - \frac{1}{2}p_1$. (Since $p_1, p_2 > \frac{2}{3}$, $1 - \frac{1}{2}p_1 < \frac{p_1+p_2}{2}$)

The automaton C recognizes the language $L_1 \cup L_2$ with interval $(\frac{2-p_1-p_2}{2}, p)$, where $p \geq \frac{1}{2}p_1$. (Again, $\frac{2-p_1-p_2}{2} < \frac{1}{2}p_1$)

Therefore by Lemma 3.6, the languages $L_1 \cap L_2$ and $L_1 \cup L_2$ are recognizable with probabilities greater than $\frac{1}{2}$. $\square$

Theorem 3.9. *The class of languages recognized by C-PRA is closed under intersection, union and complement.*

Proof. Let us consider languages L_1, L_2 recognized by some C-PRA automata. By Theorem 3.7, these languages is recognizable with probability $1-\varepsilon$, and therefore by Lemma 3.8, union and intersection of these languages are also recognizable. If a language L is recognizable by a C-PRA A, we can construct an automaton which recognizes a language $\overline{L}$ just by making accepting states of A to be rejecting, and vice versa. $\square$

Theorem 3.10. *The class of languages recognized by C-PRA is closed under inverse homomorphisms.*

Proof. Let us consider finite alphabets Σ, T, a homomorphism $h : \Sigma \longrightarrow T^*$, a language $L \subseteq T^*$ and a C-PRA $A = (Q, T, q_0, Q_F, \delta)$, which recognizes L with interval (p_1, p_2). We prove that exists an automaton $B = (Q, \Sigma, q_0, Q_F, \delta')$ which recognizes the language $h^{-1}(L)$.

Transition function δ of A sets transition matrixes V_τ, where $\tau \in T$. To determine δ', we define transition matrixes V_σ, $\sigma \in \Sigma$. Let us define a transition matrix V_{σ_k}:

$$V_{\sigma_k} = V_{[h(\sigma_k)]_m} V_{[h(\sigma_k)]_{m-1}} \ldots V_{[h(\sigma_k)]_1},$$

where $m = |h(\sigma_k)|$. Multiplication of two doubly stochastic matrixes is a doubly stochastic matrix, therefore B is a C-PRA. Automaton B recognizes $h^{-1}(L)$ with the same interval (p_1, p_2). $\square$

Corollary 3.11. *The class of languages recognized by C-PRA is closed under word quotient.*

Proof. This follows from closure under inverse homomorphisms and presence of end-markers $\looparrowright, \looparrowleft$. $\square$

We should reference that necessity of end-markers is considered in [FGK 04] and it has been proven that C-PRA automata without end-markers recognize the same class of languages as C-PRA automata with both end-markers. Thus if C-PRA without end-markers are considered, closure under word quotient remains true.

3.3.3 Recognition of $L_n = a_1^* a_2^* \dots a_n$

Theorem 3.12. *For every natural positive n, a language $L_n = a_1^* a_2^* \dots a_n^*$ is recognizable by some C-PRA with alphabet $\{a_1, a_2, \dots, a_n\}$.*

Proof. We construct a C-PRA with $n+1$ states, q_0 being the initial state, corresponding to probability distribution vector $\begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}$. The transition function is determined by $(n+1) \times (n+1)$ matrixes
$V_{a_1} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \frac{1}{n} & \dots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{1}{n} & \dots & \frac{1}{n} \end{pmatrix}$, $V_{a_2} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & \dots & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{n-1} & \dots & \frac{1}{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \frac{1}{n-1} & \dots & \frac{1}{n-1} \end{pmatrix}$, $\dots$, $V_{a_n} = \begin{pmatrix} \frac{1}{n} & \dots & \frac{1}{n} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{1}{n} & \dots & \frac{1}{n} & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$. The accepting states are $q_0 \dots q_{n-1}$, the only rejecting state is q_n. We prove, that the automaton recognizes the language L_n.

Case $\omega \in L_n$. Having read $\omega \in a_1^* a_2^* \dots a_k^*$, the automaton is in probability distribution $\begin{pmatrix} \frac{1}{k} \\ \dots \\ \frac{1}{k} \\ 0 \\ \dots \\ 0 \end{pmatrix}$. Therefore all $\omega \in L_n$ are accepted with probability 1.

Case $\omega \notin L_n$. Consider k such that $\omega = \omega_1\sigma\omega_2$, $|\omega_1| = k$, $\omega_1 \in L_n$ and $\omega_1\sigma \notin L_n$. Since all one-letter words are in L_n, $k > 0$. Let $a_t = [\omega]_k$ and $a_s = \sigma$. So we have $s < t$, $1 \leq s \leq n-1$, $2 \leq t \leq n$. Having read $\omega_1 \in a_1^*a_2^*\dots a_t^*$, the automaton is in distribution $\begin{pmatrix} \frac{1}{t} \\ \dots \\ \frac{1}{t} \\ 0 \\ \dots \\ 0 \end{pmatrix}$. After that, having read a_s, the automaton is in distribution $\begin{pmatrix} \frac{1}{s} & \dots & \frac{1}{s} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{1}{s} & \dots & \frac{1}{s} & 0 & \dots & 0 \\ 0 & \dots & 0 & \frac{1}{n-s+1} & \dots & \frac{1}{n-s+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \frac{1}{n-s+1} & \dots & \frac{1}{n-s+1} \end{pmatrix} \begin{pmatrix} \frac{1}{t} \\ \dots \\ \frac{1}{t} \\ 0 \\ \dots \\ 0 \end{pmatrix} =$
$\begin{pmatrix} \left.\begin{matrix} \frac{1}{t} \\ \dots \\ \frac{1}{t} \end{matrix}\right\}s \\ \left.\begin{matrix} \frac{t-s}{t(n-s+1)} \\ \dots \\ \frac{t-s}{t(n-s+1)} \end{matrix}\right\}n-s+1 \end{pmatrix}$. So the word $\omega_1 a_s$ is accepted with probability $1 - \frac{t-s}{t(n-s+1)}$. By Lemma 2.11, since $\frac{t-s}{t(n-s+1)} < \frac{1}{t}$, reading the symbols succeeding $\omega_1 a_s$ will not increase accepting probability. Therefore, to find maximum accepting probability for words not in L_n, we have to maximize $1 - \frac{t-s}{t(n-s+1)}$, where $s < t$, $1 \leq s \leq n-1$, $2 \leq t \leq n$. Solving this problem, we get $t = k+1, s = k$ for $n = 2k$, and we get $t = k+1, s = k$ or $t = k+2, s = k+1$ for $n = 2k+1$. So the maximum accepting probability is $1 - \frac{1}{(k+1)^2}$, if $n = 2k$, and it is $1 - \frac{1}{(k+1)(k+2)}$, if $n = 2k+1$. All in all, the automaton recognizes the language with interval $(1 - \frac{1}{\lfloor (\frac{n}{2})^2 \rfloor + n + 1},\ 1)$. (Actually, by Theorem 3.7, L_n can be recognized with probability $1 - \varepsilon$). □

Corollary 3.13. *Quantum finite automata with mixed states (model of Nayak, [N 99]) recognize $L_n = a_1^*a_2^*\dots a_n^*$ with probability $1 - \varepsilon$.*

Proof. This comes from the fact, that matrixes $V_{a_1}, V_{a_2}, \dots, V_{a_n}$ from the proof of Theorem 3.12 and the matrixes in Theorem 3.7 all have unitary prototypes (see Definition 2.12). □

3.3.4 Class of languages not recognizable by 1-way C-PRA

In this section we will prove that regular languages which minimal deterministic automaton contain certain forbidden constructions can not be recognizable by 1-way C-PRA. We start by definition of these "forbidden" constructions.

Definition 3.14. *We say that a regular language is of Type* 0 *(Figure 3.1) if the following is true for the minimal deterministic automaton recognizing this language: Exist three states q, q_1, q_2, exist words x, y such that*
1) $q_1 \neq q_2$;
2) $qx = q_1,\ qy = q_2$;
3) $q_1x = q_1,\ q_2y = q_2$;
4) $\forall t \in (x,y)^*\ \exists t_1 \in (x,y)^*\ q_1tt_1 = q_1$;
5) $\forall t \in (x,y)^*\ \exists t_2 \in (x,y)^*\ q_2tt_2 = q_2$.

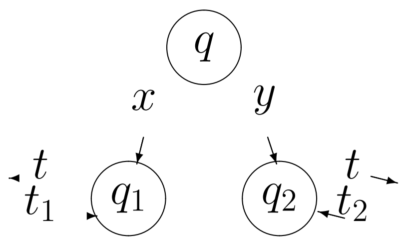

Figure 3.1: Type 0 construction

Definition 3.15. *We say that a regular language is of Type* 1 *(Figure 3.2) if the following is true for the minimal deterministic automaton recognizing this language: Exist two states* q_1*,* q_2*, exist words* x*,* y *such that*
1) $q_1 \neq q_2$;
2) $q_1 x = q_2,\ q_2 x = q_2$;
3) $q_2 y = q_1$.

Definition 3.16. *We say that a regular language is of Type* 2 *(Figure 3.3) if the following is true for the minimal deterministic automaton recognizing this language: Exist three states* q*,* q_1*,* q_2*, exist words* x*,* y *such that*
1) $q_1 \neq q_2$;
2) $qx = q_1,\ qy = q_2$;
3) $q_1 x = q_1,\ q_1 y = q_1$;
4) $q_2 x = q_2,\ q_2 y = q_2$.

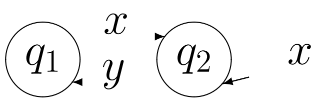

Figure 3.2: Type 1 construction

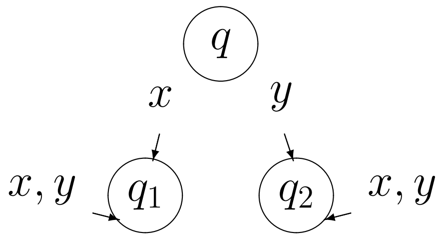

Figure 3.3: Type 2 construction

Type 1 languages are exactly those languages that violate the partial order condition of [BP 99]. Type 2 construction is more general then forbidden construction for DH-QFA considered in [AKV 00].

Lemma 3.17. *If* A *is a deterministic finite automaton with a set of states* Q *and alphabet* Σ*, then* $\forall q \in Q\ \forall x \in \Sigma^*\ \exists k > 0\ qx^k = qx^{2k}$.

Proof. We paraphrase a result from the theory of finite semigroups. Consider a state q and a word x. Since number of states is finite, $\exists m \geq 0\ \exists s \geq 1\ \forall n$ $qx^m = qx^m x^{sn}$. Take n_0, such that $sn_0 > m$. Note that $\forall t \geq 0\ qx^{m+t} = qx^{m+t}x^{sn_0}$. We take $t = sn_0 - m$, so $qx^{sn_0} = qx^{sn_0}x^{sn_0}$. Take $k = sn_0$. $\square$

Lemma 3.18. *A regular language is of Type* 0 *iff it is of Type* 1 *or Type* 2.

Proof. 1) If a language is of Type 2, it is of Type 0. Obvious.

2) If a language is of Type 1, it is of Type 0. Consider a language of Type 1 with states q_1'', q_2'' and words x'', y''. To build construction of Type 0, we

take $q = q_1 = q_1''$, $q_2 = q_2''$, $x = x''y''$, $y = x''$. That forms transitions $qx = q_1$, $qy = q_2$, $q_1x = q_1$, $q_1y = q_2$, $q_2x = q_1$, $q_2y = q_2$. We have satisfied all the rules of Type 0.

3) If a language is of Type 0, it is of Type 1 or 2. Consider a language whose minimal deterministic automaton has construction of Type 0. By Lemma 3.17,

$$\exists t \exists b \; q_1y^b = q_t \text{ and } q_ty^b = q_t;$$
$$\exists u \exists c \; q_2x^c = q_u \text{ and } q_ux^c = q_u.$$

If $q_1 \neq q_t$, by the 4th rule of Type 0, $\exists z \; q_tz = q_1$. Therefore the language is of Type 1. If $q_2 \neq q_u$, by the 5th rule of Type 0, $\exists z \; q_uz = q_2$, and the language is of Type 1.
If $q_1 = q_t$ and $q_2 = q_u$, we have $qx^c = q_1$, $qy^b = q_2$, $q_1x^c = q_1y^b = q_1$, $q_2x^c = q_2y^b = q_2$. We get the construction of Type 2 if we take $x' = x^c$, $y' = y^b$. □

The following theorem illustrates the relationship between Type 1 and Type 2 languages.

Theorem 3.19. *A regular language L is of Type 1 iff L^R is of Type 2.*

Proof. It is a well known fact, that the class of regular languages is closed under reversal.

1) Consider a Type 1 regular language $L \subset \Sigma^*$. Since L is of Type 1, it is recognized by a minimal deterministic automaton $D = (Q, \Sigma, q_0, Q_F, \delta)$ with particular two states q_1, q_2, such that $q_1 \neq q_2$, $q_1x = q_2$, $q_2x = q_2$, $q_2y = q_1$, where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0\omega = q_1$, and exists $z \in \Sigma^*$ such that $q_1z \in Q_F$ if and only if $q_2z \notin Q_F$. Minimal deterministic automata of a regular language and of its complement are isomorphic, so without loss of generality we assume that $q_1z \in Q_F$ and $q_2z \notin Q_F$.

So $\omega\{xy, x\}^*xz \subset \overline{L}$ and $\omega\{xy, x\}^*(xy)z \subset L$, and in the case of the reverse of L, $z^Rx^R\{y^Rx^R, x^R\}^*\omega^R \subset \overline{L^R}$ and $z^R(y^Rx^R)\{y^Rx^R, x^R\}^*\omega^R \subset L^R$. We denote $\sigma_1 = x^R$, $\sigma_2 = y^Rx^R$, hence $z^R\sigma_1\{\sigma_2, \sigma_1\}^*\omega^R \subset \overline{L^R}$ and $z^R\sigma_2\{\sigma_2, \sigma_1\}^*\omega^R \subset L^R$.

Consider a minimal deterministic automaton $D^R = (Q^R, \Sigma, s_0, Q_F^R, \delta^R)$, which recognizes L^R. Let $s = s_0z^R$. Let $Q_1 = \{s\tau \mid \tau \in \sigma_1\{\sigma_2, \sigma_1\}^*\}$ and $Q_2 = \{s\tau \mid \tau \in \sigma_2\{\sigma_2, \sigma_1\}^*\}$. For any $q \in Q_1$, $q\omega^R \notin Q_F^R$ and for any $q \in Q_2$, $q\omega^R \in Q_F^R$. Therefore $Q_1 \cap Q_2 = \emptyset$. Furthermore, it is impossible to go from a state in Q_1 to a state in Q_2, or vice versa, using only words in $\{\sigma_1, \sigma_2\}^*$. So $s \notin Q_1$ and $s \notin Q_2$.

Consider a relation $R = \{(s_i, s_j) \in Q_1^2 \mid s_j \in s_i\{\sigma_1, \sigma_2\}^*\}$. R is a weak ordering, so $R' = \{(s_i, s_j) \mid s_iRs_j \text{ and } s_jRs_i\}$ is an equivalence relation, partitioning Q_1 into equivalence classes. Since the number of states in Q_1 is finite, exists a class $S \subset Q_1$, which is minimal, i.e, $\forall q \in S\ \forall \tau \in \{\sigma_1, \sigma_2\}^*$ $q\tau \in S$. Since $S \subset Q_1$, exists a word $\tau_1 \in \{\sigma_1, \sigma_2\}^*$, such that $s(\sigma_1\tau_1) \in S$. Now by Lemma 3.17, $\exists p > 0\ \exists s_1 \in S\ s(\sigma_1\tau_1)^p = s_1$ and $s_1(\sigma_1\tau_1)^p = s_1$. Since S is an equivalence class of R', $\forall q \in S\ \forall \tau \in \{\sigma_1, \sigma_2\}^*\ \exists \tau_2 \in \{\sigma_1, \sigma_2\}^*$ $q(\tau\tau_2) = q$. So, exists τ_2, such that $s_1(\sigma_2\tau_2) = s_1$.

Let us denote $\alpha = (\sigma_1\tau_1)^p$, $\beta = \sigma_2\tau_2$, so $s\alpha = s_1$, $s_1\alpha = s_1$, $s_1\beta = s_1$, where $s_1 \in Q_1$.

By Lemma 3.17, it is possible to construct a sequence of states t_0, t_1,..., t_{m-1},..., where $t_0 = s$, such that

$t_0(\beta\alpha^{k_1}) = t_1$ and $t_1\alpha^{k_1} = t_1$,

$t_1(\beta\alpha^{k_2}) = t_2$ and $t_2\alpha^{k_2} = t_2$,

...

$t_{m-1}(\beta\alpha^{k_m}) = t_m$ and $t_m\alpha^{k_m} = t_m$,

...

Because $\beta \in \sigma_2\{\sigma_1, \sigma_2\}^*$ and $\alpha \in \sigma_1\{\sigma_1, \sigma_2\}^*$, $\forall i > 0\ t_i \in Q_2$. Let $T_m = \{t_0, \dots, t_m\}$. Since the number of states in Q_2 is finite, exists i, such that $t_i \in T_{i-1}$. So, exists j, $0 < j < i$, such that $t_j = t_i$ and starting with t_j, the sequence becomes periodic. Let $k = k_1k_2 \dots k_i$. Now, $\forall m \geq 0\ t_m(\beta\alpha^k) = t_{m+1}$ and $t_{m+1}\alpha^k = t_{m+1}$. By Lemma 3.17, $\exists r > 0\ \exists s_2$, such that $s(\beta\alpha^k)^r = s_2$ and $s_2(\beta\alpha^k)^r = s_2$. The state $s_2 = t_r$, so $s_2 \in Q_2$ and $s_2\alpha^k = s_2$.

So we have $s\alpha^k = s_1$, $s_1\alpha^k = s_1$, $s_1(\beta\alpha^k)^r = s_1$, $s(\beta\alpha^k)^r = s_2$, $s_2(\beta\alpha^k)^r = s_2$, $s_2\alpha^k = s_2$. Since $s_1 \in Q_1$, $s_2 \in Q_2$, s_1 is not equal to s_2, thus we have obtained a Type 2 construction.

2) Consider a Type 2 regular language $L \subset \Sigma^*$. Since L is of Type 2, it is recognized by a minimal deterministic automaton $D = (Q, \Sigma, q_0, Q_F, \delta)$ with particular three states q, q_1, q_2, such that $q_1 \neq q_2$, $qx = q_1$, $q_1x = q_1$, $q_1y = q_1$, $qy = q_2$, $q_2x = q_2$, $q_2y = q_2$, where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0\omega = q$, and exists $z \in \Sigma^*$ such that $q_1z \in Q_F$ if and only if $q_2z \notin Q_F$. Without loss of generality we assume that $q_1z \in Q_F$ and $q_2z \notin Q_F$.

So $\omega x\{x, y\}^*z \subset L$ and $\omega y\{x, y\}^*z \subset \overline{L}$, and in the case of the reverse of L, $z^R\{x^R, y^R\}^*x^R\omega^R \subset L^R$ and $z^R\{x^R, y^R\}^*y^R\omega^R \subset \overline{L^R}$. We denote $\sigma_1 = x^R$, $\sigma_2 = y^R$, hence $z^R\{\sigma_1, \sigma_2\}^*\sigma_1\omega^R \subset L^R$ and $z^R\{\sigma_1, \sigma_2\}^*\sigma_2\omega^R \subset \overline{L^R}$.

Consider a minimal deterministic automaton $D^R = (Q^R, \Sigma, s_0, Q_F^R, \delta^R)$, which recognizes L^R. Let $s = s_0z^R$. Let $Q_1 = \{s\tau \mid \tau \in \{\sigma_1, \sigma_2\}^*\sigma_1\}$ and $Q_2 = \{s\tau \mid \tau \in \{\sigma_1, \sigma_2\}^*\sigma_2\}$. For any $t \in Q_1$, $t\omega^R \in Q_F^R$ and for any $t \in Q_2$, $t\omega^R \notin Q_F^R$. Therefore $Q_1 \cap Q_2 = \emptyset$.

Let $T = Q_1 \cup Q_2$. Consider a relation $R = \{(s_i, s_j) \in T^2 \mid s_j \in s_i\{\sigma_1, \sigma_2\}^*\}$. R is a weak ordering, so $R' = \{(s_i, s_j) \mid s_iRs_j$ and $s_jRs_i\}$ is an equivalence relation, partitioning T into equivalence classes. Since the number of states in T is finite, exists a class $S \subset T$, which is minimal, i.e, $\forall t \in S\ \forall \tau \in \{\sigma_1, \sigma_2\}^*\ t\tau \in S$.

Consider a state $t \in S$. If the state t is in Q_1 then $t\sigma_2 \in S$ is in Q_2. If the state t is in Q_2 then $t\sigma_1 \in S$ is in Q_1. So exist t_1, t_2, such that $t_1 \in Q_1 \cap S$, $t_2 \in Q_2 \cap S$. Take $s_1 \in Q_1 \cap S$. By Lemma 3.17, $\exists k > 0\ \exists s_2$, such that $s_1\sigma_2^k = s_2$ and $s_2\sigma_2^k = s_2$. The state s_2 is in $Q_2 \cap S$. Since S is an equivalence class of R', $\exists \sigma \in \{\sigma_1, \sigma_2\}^*$, such that $s_2\sigma = s_1$.

So we have $s_1\sigma_2^k = s_2$, $s_2\sigma_2^k = s_2$, $s_2\sigma = s_1$. Since $s_1 \in Q_1$, $s_2 \in Q_2$, s_1 is not equal to s_2, thus we have obtained a Type 1 construction. $\square$

Remark 3.20. Both C-DRA and C-QFA-P (see Section 3.5) recognize exactly the regular languages for which the corresponding minimal deterministic finite automata do not contain the following construction ([HS 66, BP 99]), denoted henceforth as Type A construction (Figure 3.4): Exist two states q_1, q_2, exist words x, y such that

1) $q_1 \neq q_2$;

2) $q_1x = q_2,\ q_2x = q_2$.

Similarly as in Theorem 3.19, we can demonstrate that a regular language L is of Type A if and only if the language L^R is of Type A.

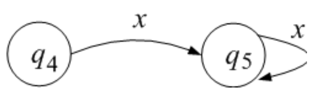

Figure 3.4: Type A construction

Finally we prove that every language of Type 0 is not recognizable by any C-PRA.

Definition 3.21. *By* $q \xrightarrow{S} q'$, $S \subset \Sigma^*$, *we denote that there is a positive probability to get to a state* q' *by reading a single word* $\xi \in S$, *starting in a state* q.

Lemma 3.22. *If a regular language is of Type* 2, *it is not recognizable by any C-PRA.*

Proof. Assume from the contrary, that A is a C-PRA automaton which recognizes a language $L \subset \Sigma^*$ of Type 2.

Since L is of Type 2, it is recognized by a minimal deterministic automaton D with particular three states q, q_1, q_2 such that $q_1 \neq q_2$, $qx = q_1$, $qy = q_2$, $q_1x = q_1$, $q_1y = q_1$, $q_2x = q_2$, $q_2y = q_2$, where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0\omega = q$, where q_0 is an initial state of D, and exists a word $z \in \Sigma^*$, such that $q_1z = q_{acc}$ if and only if $q_2z = q_{rej}$, where

q_{acc} is an accepting state and q_{rej} is a rejecting state of D. Without loss of generality we assume that $q_1 z = q_{acc}$ and $q_2 z = q_{rej}$.

The transition function of the automaton A is determined by doubly stochastic matrixes $V_{\sigma_1}, \ldots, V_{\sigma_n}$. The words from the construction of Type 2 are $x = \sigma_{i_1} \ldots \sigma_{i_k}$ and $y = \sigma_{j_1} \ldots \sigma_{j_s}$. The transitions induced by words x and y are determined by doubly stochastic matrixes $X = V_{\sigma_{i_k}} \ldots V_{\sigma_{i_1}}$ and $Y = V_{\sigma_{j_s}} \ldots V_{\sigma_{j_1}}$. Similarly, the transitions induced by words ω and z are determined by doubly stochastic matrixes W and Z. By Corollary 2.30, exists $K > 0$, such that

$$\forall i \ (X^K)_{i,i} > 0 \text{ and } (Y^K)_{i,i} > 0. \tag{3.8}$$

Consider a relation between the states of the automaton defined as $R = \{(q_i, q_j) \mid q_i \overset{(x^K,y^K)^*}{\longrightarrow} q_j\}$. By (3.8), this relation is reflexive.

Suppose exists a word $\xi = \xi_1 \xi_2 \ldots \xi_k$, $\xi_s \in \{x^K, y^K\}$, such that $q \xrightarrow{\xi} q'$. This means that $q \xrightarrow{\xi_1} q_{i_1}$, $q_{i_1} \xrightarrow{\xi_2} q_{i_2}, \ldots, q_{i_{k-1}} \xrightarrow{\xi_k} q'$. By Corollary 2.32, since both X^K and Y^K are doubly stochastic, $\exists \xi'_k \ldots \xi'_1$, $\xi'_s \in \{(x^K)^*, (y^K)^*\}$, such that $q' \xrightarrow{\xi'_k} q_{i_{k-1}}, \ldots, q_{i_2} \xrightarrow{\xi'_2} q_{i_1}$, $q_{i_1} \xrightarrow{\xi'_1} q$, therefore $q' \xrightarrow{\xi'} q$, where $\xi' \in (x^K, y^K)^*$. So the relation R is symmetric.

Surely R is transitive. Therefore all states of A may be partitioned into equivalence classes $[q_0], [q_{i_1}], \ldots, [q_{i_n}]$. Let us renumber the states of A in such a way, that states from one equivalence class have consecutive numbers. First come the states in $[q_0]$, then in $[q_{i_1}]$, etc.

Consider the word $x^K y^K$. The transition induced by this word is determined by a doubly stochastic matrix $C = Y^K X^K$. We prove the following proposition. States q_a and q_b are in one equivalence class if and only if $q_a \to q_b$ with matrix C. Suppose $q_a \to q_b$. Then $(q_a, q_b) \in R$, and q_a, q_b are in one equivalence class. Suppose q_a, q_b are in one equivalence class. Then

$$q_a \xrightarrow{\xi_1} q_{i_1}, q_{i_1} \xrightarrow{\xi_2} q_{i_2}, \ldots, q_{i_{k-1}} \xrightarrow{\xi_k} q_b, \text{ where } \xi_s \in \{x^K, y^K\}. \tag{3.9}$$

By (3.8), $q_i \xrightarrow{x^K} q_i$ and $q_j \xrightarrow{y^K} q_j$. Therefore, if $q_i \xrightarrow{x^K} q_j$, then $q_i \xrightarrow{x^K y^K} q_j$, and again, if $q_i \xrightarrow{y^K} q_j$, then $q_i \xrightarrow{x^K y^K} q_j$. That transforms (3.9) to

$$q_a \xrightarrow{(x^K y^K)^t} q_b, \text{ where } t > 0. \tag{3.10}$$

We have proved the proposition.

By the proved proposition, due to the renumbering of states, matrix C is a block diagonal matrix, where each block corresponds to an equivalence class of the relation R. Let us identify these blocks as $C_0, C_1, \ldots, C_n$. By

(3.8), a Markov chain with matrix C is aperiodic. Therefore each block C_r corresponds to an aperiodic irreducible doubly stochastic Markov chain with states $[q_{i_r}]$. By Corollary 2.28, $\lim_{m\to\infty} C^m = J$, J is a block diagonal matrix, where for each $(p\times p)$ block C_r $(C_r)_{i,j} = \frac{1}{p}$. Relation $q_i \xrightarrow{(y^K)^*} q_j$ is a subrelation of R, therefore Y^K is a block diagonal matrix with the same block ordering and sizes as C and J. (This does not eliminate possibility that some block of Y^K is constituted of smaller blocks, however.) Therefore $JY^K = J$, and $\lim_{m\to\infty} Z(Y^K X^K)^m W = \lim_{m\to\infty} Z(Y^K X^K)^m Y^K W = ZJW$. So

$$\forall\varepsilon > 0\ \exists m\ \left\|\Big(Z(Y^K X^K)^m W - Z(Y^K X^K)^m Y^K W\Big)Q_0\right\| < \varepsilon. \tag{3.11}$$

However, by construction of Type 2, $\forall k\ \forall m\ \omega(x^k y^k)^m z \in L$ and $\omega y^k(x^k y^k)^m z \notin L$. This requires existence of $\varepsilon > 0$, such that

$$\forall m\ \left\|\Big(Z(Y^K X^K)^m W - Z(Y^K X^K)^m Y^K W\Big)Q_0\right\| > \varepsilon. \tag{3.12}$$

This is a contradiction. □

Lemma 3.23. *If a regular language is of Type 1, it is not recognizable by any C-PRA.*

Proof. Proof is nearly identical to that of Lemma 3.22.

Consider a C-PRA which recognizes the language L of Type 1. We prove that for words x, y exists constant K, such that for every ε exists m, such that for two words $\xi_1 = \omega(x^K(xy)^K)^m z$ and $\xi_2 = \omega(x^K(xy)^K)^m x^K z$, $|p_{\xi_1} - p_{\xi_2}| < \varepsilon$.

The transition function of the automaton A is determined by doubly stochastic matrixes $V_{\sigma_1}, \dots, V_{\sigma_n}$. The words from the construction of Type 2 are $x = \sigma_{i_1}\dots\sigma_{i_k}$ and $y = \sigma_{j_1}\dots\sigma_{j_s}$. The transitions induced by words x and y are determined by doubly stochastic matrixes $X = V_{\sigma_{i_k}}\dots V_{\sigma_{i_1}}$ and $Y = V_{\sigma_{j_s}}\dots V_{\sigma_{j_1}}$. Similarly, the transitions induced by words ω and z are determined by doubly stochastic matrixes W and Z.

By Corollary 2.30 we can select such K that

$$\forall i\ (X^K)_{i,i} > 0 \text{ and } (Y^K)_{i,i} > 0. \tag{3.13}$$

.

Matrix $C = (YX)^K X^K$ corresponds to reading of $x^K(xy)^K$. We consider a relation between the states of the automaton defined as $R = \{(q_i, q_j) \mid q_i \xrightarrow{(x^K(xy)^K)^*} q_j\}$. This relation by Corollary 2.34 divides states into equivalence classes.

$q \xrightarrow{(x^K)^*} q'$ is subrelation of R. To show that rewrite $q \xrightarrow{(x^K)^*} q'$ as sequence $q \xrightarrow{x^K} q_{i_1}, \ldots, q_{i_{k-2}} \xrightarrow{x^K} q_{i_{k-1}}, q_{i_{k-1}} \xrightarrow{x^K} q'$. As K selected so that $q_j \xrightarrow{(xy)^K} q_j$ for any j then we can substitute x^K with $x^K(xy)^K$ at each step $q \xrightarrow{x^K(xy)^K} q_{i_1}, \ldots,$ $q_{i_{k-2}} \xrightarrow{x^K(xy)^K} q_{i_{k-1}}, q_{i_{k-1}} \xrightarrow{x^K(xy)^K} q'$, getting $q \xrightarrow{(x^K(xy)^K)^*} q'$. Thus q_i and q_j are in one equivalence class in respect to R.

Due to the renumbering of states, matrix C is a block diagonal matrix, where each block corresponds to an equivalence class of the relation R. Let us identify these blocks as $C_0, C_1, \ldots, C_n$. By (3.13), a Markov chain with matrix C is aperiodic. Therefore each block C_r corresponds to an aperiodic irreducible doubly stochastic Markov chain with states $[q_{i_r}]$. By Corollary 2.28, $\lim\limits_{m\to\infty} C^m = J$, J is a block diagonal matrix, where for each $(p \times p)$ block C_r $(C_r)_{i,j} = \frac{1}{p}$. As relation $q_i \xrightarrow{(x^K)^*} q_j$ is a subrelation of R, therefore X^K is a block diagonal matrix with the same block ordering and sizes as C and J. (This does not eliminate possibility that some block of X^K is constituted of smaller blocks, however.) Therefore $JX^K = J$, and $\lim\limits_{m\to\infty} ZX^K((YX)^K X^K)^m W = \lim\limits_{m\to\infty} Z((YX)^K X^K)^m W = ZJW$.

So

$$\forall \varepsilon > 0\ \exists m\ \left\| \Big(Z(X^K((YX)^K X^K)^m W - Z((YX)^K X^K)^m W \Big) Q_0 \right\| < \varepsilon. \tag{3.14}$$

However, by construction of Type 1, we can select z such that $\omega(x^k(xy)^k)^m x^K z \in L$ and $\omega(x^k(xy)^k))^m z \notin L$. This requires existence of $\varepsilon > 0$, such that

$$\forall m\ \left\| \Big(ZX^K((YX)^K X^K)^m W - Z((YX)^K X^K)^m W \Big) Q_0 \right\| > \varepsilon. \tag{3.15}$$

This is a contradiction.

□

Theorem 3.24. *If a regular language is of Type* 0*, it is not recognizable by any C-PRA.*

Proof. By Lemmas 3.18, 3.22, 3.23. □

We proved (Lemma 3.18) that the construction of Type 0 is a generalization the construction proposed by [BP 99]. Also it can be easily noticed, that the Type 0 construction is a generalization of construction proposed by [AKV 00]. (Constructions of [BP 99] and [AKV 00] characterize languages, not recognized by measure-many quantum finite automata of [KW 97].)

Corollary 3.25. *Languages (a,b)*a and a(a,b)* are not recognized by C-PRA.*

Proof. Both languages are of Type 0. □

Corollary 3.26. *Class of languages recognizable by C-PRA is not closed under homomorphisms.*

Proof. Consider a homomorphism $a \to a$, $b \to b$, $c \to a$. Similarly as in Theorem 3.12, the language (a,b)*cc* is recognizable by a C-PRA. (Take $n = 2$, $V_a = V_{a_1}$, $V_b = V_{a_1}$, $V_c = V_{a_2}$ from Theorem 3.12, $Q_F = \{q_1\}$) However, by Corollary 3.25 the language (a,b)*aa*=(a,b)*a is not recognizable. □

3.4 1-way Probabilistic Reversible DH Automata

3.4.1 Definition

Taken the definition 3.1 of Probabilistic Reversible automata we define word acceptance as specified in Definition 2.38. The set of accepting states is Q_A and set of rejecting states is Q_R, these states are halting. We define language recognition as in Definition 2.42. That completes formal definition of 1-way DH-PRA automata.

The 1-way DH-PRA can be viewed alternatively as the automaton with classical acceptance, but with different form of transition matrix. Instead of halting once reaching the halting state we can consider that automaton continues to read input till the end of but remain in the same halting state. In this case transition matrixes V_σ for some σ are not doubly stochastic, but of the following form. As we can enumerate states of DH-PRA for V_σ in such way that:

1. q_1 ... q_k are states, from which halting states are not accessible,
2. q_{k+1} ... q_{n-l} are non-halting states from which halting states are accessible,
3. q_{n-l+1} ... q_n are halting states,

and transition matrix V_σ will look so: $\begin{array}{r} k\{ \\ \\ l\{ \end{array} \begin{pmatrix} DST & O & O \\ O & a_{ij} & O \\ O & a_{ij} & I \end{pmatrix}$, where

- DST - doubly stochastic matrix,

- I - unit matrix,
- $\forall k+1 \leq j \leq n-l : \sum\limits_{i=1}^{n} \alpha_{ij} = 1$ (it's still stochastic),
- $\forall k+1 \leq i \leq n-l : \sum\limits_{j=1}^{n} \alpha_{ij} <= 1$ (as originated from double stochastic matrix where sum in each row is one).

According to definitions 2.17 states q_{k+1} ... q_{n-l} are transient and states q_1 ... q_k and q_{n-l+1} ... q_n are recurrent, with q_{n-l+1} ... q_n being absorbing for Markov chain induced by transitions V_σ. Note that for different letters of the alphabet σ the numbering of non halting states will be different.

We call matrix of such type DH-stochastic matrix. Certainly transformation that corresponds to the reading of a sequence of letters also is described by a DH-stochastic matrix.

Lemma 3.27. *For any $\sigma_s, \sigma_t \in \Sigma$: $V_{\sigma_s} \cdot V_{\sigma_t}$ - is also DH-stochastic matrix.*

Proof. Follows from the matrix manipulation. To show that for states from which halting states are not accessible the matrix is double stochastic observe that no sum in the row can exceed 1 still and also can not be less then 1 as otherwise summing by rows and columns would give different results. □

It should be noted however that transient states in $V_{\sigma_s} \cdot V_{\sigma_t}$ could be different from transient states in $V_{\sigma_s} \cdot$ and V_{σ_t}.

To prove forbidden constructions for DH-PRA we need to consider the behavior of the Markov chain induced by transition V_{σ_k} in long run.

Lemma 3.28. *There exists such K that* $\lim\limits_{n\to\infty} A^{Kn} = \begin{pmatrix} k\{DST' & O & O \\ O & O & O \\ l\{O & a_{ij} & I \end{pmatrix}.$
where DST' is block diagonal matrix with each block being double stochastic matrix $\frac{1}{k_i}$ with k_i size of block.

Proof. Follows if taken K such that $A^K_{i,i} > 0$ for all recurrent states (possible by Lemma 2.29). By Theorem 2.26 there is 0 probability to be at the transient state. As recurrent states in A^K form doubly stochastic matrix then they can be split into equivalence classes in respect to communication property (see Corollary 2.34) and each block diagonal submatrix corresponds to states in one equivalence classes. The values in these submatrixes are determined by Corollary 2.28 . □

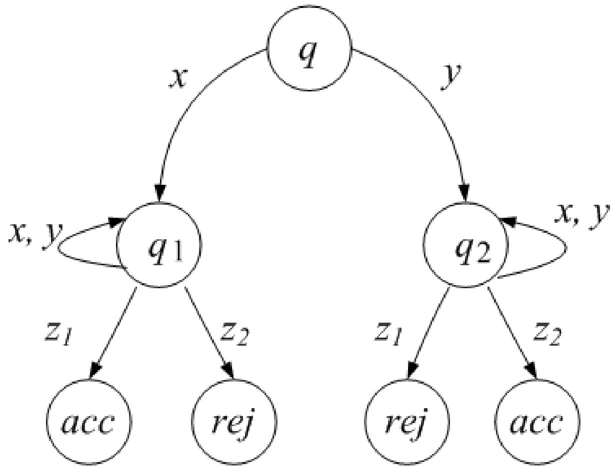

Figure 3.5: Type 3 construction

3.4.2 Class of languages non recognizable by 1-way DH-PRA

It is easy to see that class of languages recognized by C-PRA is a proper subclass of languages recognized by DH-PRA.

Example 3.29. The language a(a,b)* known not to be recognizable by C-PRA is recognizable by DH-PRA.

In this section we will prove that regular languages which minimal deterministic automaton contain certain forbidden constructions can not be recognizable by 1-way DH-PRA. We start by definition of these forbidden constructions, that are quite similar to ones defined for C-PRA.

First class of languages to be considered is Type 1 described in C-PRA section (see Figure 3.2).

Second class is modification of Type 2.

Definition 3.30. *We say that a regular language is of Type* 3 *(Figure 3.5) if a regular language is of Type* 2 *and additional conditions hold for states* q_1, q_2*: there exist 2 words* z_1 *and* z_2 *such that*

1. *reading of* z_1 *when being in* q_1 *leads to accepting state and reading* z_1 *when being in* q_2 *leads to not accepting state;*

2. *reading of* z_2 *when being in* q_2 *leads to accepting state and reading* z_2 *when being in* q_1 *leads to not accepting state.*

Theorem 3.31. *If a regular language is of Type* 1*, it is not recognizable by any PRA-DH.*

Proof. Assume from the contrary, that A is a PRA-DH automaton which recognizes a language $L \subset \Sigma^*$ of Type 1.

Since L is of Type 1, it is recognized by a deterministic automaton D which has two states q_1, q_2 such that $q_1 \neq q_2$, $q_1 x = q_2$, $q_2 y = q_1$, $q_2 x = q_2$ where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0 \omega = q_1$, where q_0 is an initial state of D, and exists a word $z \in \Sigma^*$, such that $q_1 z = q_{acc}$ if and only if $q_2 z = q_{rej}$, where q_{acc} is an accepting state and q_{rej} is a rejecting state of D.

The transition function of the automaton A is determined by doubly stochastic matrices $V_{\sigma_1}, \ldots, V_{\sigma_n}$. The words $x = \sigma_{i_1} \ldots \sigma_{i_k}$ and $y = \sigma_{j_1} \ldots \sigma_{j_s}$, the transitions induced by words x and y are determined by doubly stochastic matrices $X = V_{\sigma_{i_k}} \ldots V_{\sigma_{i_1}}$ and $Y = V_{\sigma_{j_s}} \ldots V_{\sigma_{j_1}}$. Similarly, the transitions induced by word ω is determined by doubly stochastic matrix W.

Let us select 2 words x_1 and x_2 of the form $x_1 = \omega(x^K(xy)^K)^m$ and $x_2 = \omega(x^K(xy)^K)^m x^K$.

We will show that for any ε we can select K and m such that $|p_{x_1} - p_{x_2}| < \varepsilon$. Then as $x_1 z \in L$ and $x_2 z \notin L$ we get a contradiction.

We take K to be

- $K > n$ where n is number of states of given DH-PRA A;
- K is multiple of $K_1 * n$ such that $(X^{K_1})_{i,i} > 0$ for all non-halting states of A recurrent in respect to X;
- K is multiple of $K_2 * n$ such that $((YX)^{K_2})_{i,i} > 0$ for all non halting states of A recurrent in respect to YX.

We can select such K_1 and K_2 by Corollary 2.30. We should note however that recurrent states in X and YX in general could be different! Given $K > n$ we get that any transient state for X^K is also transient state for $(YX)^K X^K$. As for any transient state q of any DH stochastic matrix A some absorbing state will be accessible by A^K $K \geq n$ in 1 step, and if q' is absorbing state that $q \xrightarrow{x^K} q'$, and $q' \xrightarrow{(xy)^K} q'$.

But there could be some states recurrent for X^K that are transient states for $(YX)^K X^K$. However if q_i and q_j are states recurrent for X^K and $q_i \leftrightarrow q_j$ for X^K and q_i is transient for $(YX)^K X^K$ then q_j is transient for $(YX)^K X^K$ as well.

q_i transient in respect to $(YX)^K X^K$ means there is some sequence of letters starting with x^K that leads to the absorbing state from q_i but not

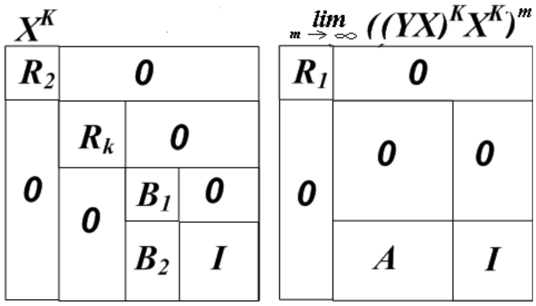

Figure 3.6: The structure of the matrixes X^K and $\lim\limits_{m\to\infty}((YX)^K X^K)^m$

from q_j. But that is contradiction as for any $q\prime$ $q_i \xrightarrow{X^K} q'$ will also hold $q_j \xrightarrow{X^K} q'$ as we selected K to satisfy conditions of Lemma 2.33.

So for $\lim\limits_{m\to\infty}$ we get $((YX)^K X^K)^m$ converges to some matrix J of the form described in Lemma 3.28. $X^K J = J$ follows from matrix multiplication rules: X^K in respect to non-halting recurrent states of $((YX)^K X^K)$ is a block diagonal matrix of the same block ordering and size (although it is possible that some of blocks consist of smaller blocks), but for transient and halting states there is the same position and size of identity matrix and all the rows corresponding to transient states in J are 0 rows (see also Figure 3.6).

That means that after reading $x_1 = \omega(x^K(xy)^K)^m$ and $x_2 = \omega(x^K(xy)^K)^m x^K$ we will get arbitrary close probability distributions that gives us required contradiction. Or formally
$\lim\limits_{m\to\infty} ZX^K((YX)^K X^K)^m W = \lim\limits_{m\to\infty} Z((YX)^K X^K)^m W = ZJW$. So

$$\forall\varepsilon > 0\ \exists m\ \left\|\Big(Z(X^K((YX)^K X^K)^m W - Z((YX)^K X^K)^m W\Big)Q_0\right\| < \varepsilon. \tag{3.16}$$

As we can select z such that $\omega(x^k(xy)^k)^m x^K z \in L$ and $\omega(x^k(xy)^k))^m z \notin L$, that requires existence of $\varepsilon > 0$, such that

$$\forall m\ \left\|\Big(ZX^K((YX)^K X^K)^m W - Z((YX)^K X^K)^m W\Big)Q_0\right\| > \varepsilon. \tag{3.17}$$

□

Theorem 3.32. *If a regular language is of Type* 3 *then it is not recognizable by any PRA-DH.*

Proof. Assume from the contrary, that A is a PRA-DH automaton which recognizes a language $L \subset \Sigma^*$ of Type 3.

Since L is of Type 3, it is recognized by a minimal deterministic automaton D with particular three states q, q_1, q_2 such that $q_1 \neq q_2$, $qx = q_1$, $qy = q_2$, $q_1x = q_1$, $q_1y = q_1$, $q_2x = q_2$, $q_2y = q_2$, where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0\omega = q$, where q_0 is an initial state of D, and exist words $z_1 \in \Sigma^*$ $z_2 \in \Sigma^*$, such that $q_1z_1 = q_{acc}$ and $q_1z_2 = q_{rej}$, $q_2z_1 = q_{rej}$ and $q_2z_2 = q_{acc}$, where q_{acc} is an accepting state and q_{rej} is a rejecting state of D.

The transition function of the automaton A is determined by doubly stochastic matrices $V_{\sigma_1}, \dots, V_{\sigma_n}$. The words from the construction of Type 3 are $x = \sigma_{i_1} \dots \sigma_{i_k}$ and $y = \sigma_{j_1} \dots \sigma_{j_s}$. The transitions induced by words x and y are determined by doubly stochastic matrices $X = V_{\sigma_{i_k}} \dots V_{\sigma_{i_1}}$ and $Y = V_{\sigma_{j_s}} \dots V_{\sigma_{j_1}}$. Similarly, the transitions induced by words ω and z_1 z_2 are determined by doubly stochastic matrices W and Z_1 and Z_2.

Let us select 2 words x_1 and x_2 of the form $x_1 = \omega(y^K(x^Ky^K)^m$ and $x_2 = \omega(x^Ky^K)^m$.

We take K to be

- $K > n$ where n is number of states of given PRA-DH A
- K is multiple of $K_1 * n$ where $(Y^{K_1})_{i,i} > 0$ for all non-halting states of A recurrent in respect to X;
- K is multiple of $K_2 * n$ where $(Y^{K_2})_{i,i} > 0$ for all non-halting states in A recurrent in respect to Y^{K_2}

We can select such K_1 and K_2 by Corollary 2.30. Recurrent states in X and Y in general could be different, by the selection of $K > n$ any transient state in Y^K is also transient in Y^KX^K. q_i transient in respect to Y^K means some halting state is accessible in 1 step from q_i due to selection of $K > n$, then q_i either
a) transient in respect to X^K and then some transient state is accessible in 1 step with X^K, thus q_i transient in respect to Y^KX^K or
b) q_i is recurrent in respect to X^K and then $q_i \xrightarrow{X^K} q_i$ but then q_i transient in respect to Y^KX^K It easy to see that for any transient state q of any DH stochastic matrix A some absorbing state will be accessible by A^K $K \geq n$ in 1 step. (As $q \xrightarrow{x^K} q'$ where q' is absorbing state, and $q' \xrightarrow{(xy)^K} q'$.)

There could be some states recurrent for Y^K that are transient states for Y^KX^K. However if q_i and q_j are states recurrent for Y^K and $q_i \leftrightarrow q_j$ and q_i is transient for Y^KX^K then q_j is transient for Y^KX^K as well. That holds as we selected K such to satisfy conditions of Lemma 2.33 for both X and Y. Then assume from contrary q_j is recurrent for Y^KX^K, if
a) q_j transient in respect to X^K then some halting state is accessed in 1 step

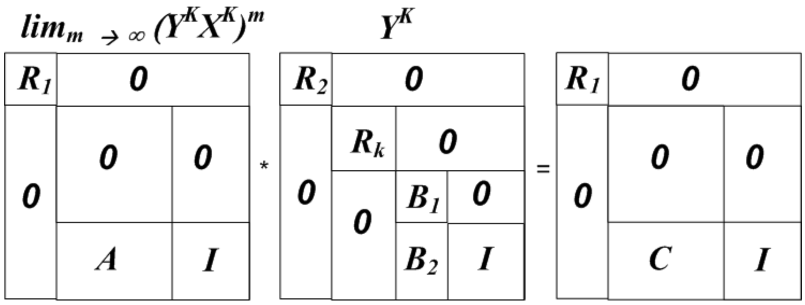

Figure 3.7: The structure of the matrixes Y^K and $\lim_{m\to\infty}(Y^K X^K)^m$ and $\lim_{m\to\infty}(Y^K X^K)^m \times Y^K$

in respect to X^K and thus q_j transient in respect to $Y^K X^K$
b) q_j recurrent in respect to X^K, then $q_j \xrightarrow{X^K} q_j$ and as $q_j \xrightarrow{X^K} q_j$ and then again q_j transient in respect to $Y^K X^K$

So for $\lim_{m\to\infty}$ we get $(Y^K X^K)^m$ converges to some matrix J of the form described in Lemma 3.28. Consider JY^K. In respect to non-halting recurrent states of $(Y^K X^K)$ the corresponding submatrix of Y^K is a block diagonal matrix of the same block ordering and size (although it is possible that some of blocks consist of smaller blocks). For transient and halting states there is the same position and size of identity matrix , the all 0 rows corresponding to transient states in J remain 0 rows, but rows corresponding to halting states are changed (see also Figure 3.7)

That means that after reading $x_1 = \omega y^K (x^K y^K)^m$ and $x_2 = \omega (x^K y^K)^m$ from whatever starting state we will get arbitrary close probability distributions for non-halting states, but different probability distributions for the halting states.

Then consider reading z_1 from such probability distribution, we receive that the distribution over the absorbing states after reading ωx and ωy to be added to the same distribution after reading remaining part of the word. That leads that after ωx accepting probability should be more then after reading ωy. If we observe z_2 we receive contradiction as now ωx accepting probability should be less then after reading ωy. That leads to contradiction. □

3.4.3 Closure properties

In this section we prove that the class of languages recognizable by DH-PRA automata is not closed by the union. In [AKV 00] there is proposed a lan-

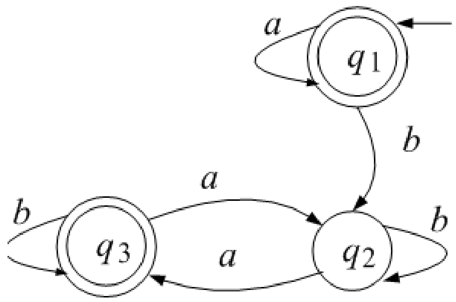

Figure 3.8: Minimal Automaton of L_1

guage not recognisable by DH-QFA which is union of languages recognizable by DH-QFA, we basically follow their proof. Although forbidden construction for DH-QFA considered in [AKV 00] is different from considered above we find also Type 3 forbidden construction in this language.

Theorem 3.33. *There are two languages L_2 and L_3 which are recognizable by DH-PRA, but the union of them $L_1 = L_2 \cup L_3$ is not recognizable by DH-PRA.*

Proof. Let L_1 be the language consisting of all words that start with any number of letters a and after first letter b (if there is one) there is an odd number of letters a. Its minimal automaton G_1 is shown in Fig. 3.8.

This language satisfies the conditions of Theorem 3.32 q_1, q_2 and q_3 of Theorem 3.32 are just q_1, q_2 and q_3 of G_1. x, y, z_1, z_2 are b, aba, ab and b. Hence it cannot be recognized by a DH-PRA. Consider two other languages L_2 and L_3 defined as follows. L_2 consists of all words which start with an even number of letters a and after firs letter b (if there is one) there is an odd number of letters a. L_3 consists of all wards which start with an odd number of letters a and after firs letter b (if there is one) there is an odd number of letters a. It is easy to see that $L_1 = L_2 \cup L_3$. The minimal automatons G_2 and G_3 are shown on Fig. 3.9 and Fig. 3.10.

We construct two DH-PRA automata K_2 and K_3 which recognize languages G_2 and G_3. The automaton K_2 consists of 12 states: q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11} and q_{12}, where $Q_{non} = \{q_1, q_2, q_3, q_4, q_{12}\}$, $Q_{rej} = \{q_5, q_6, q_7, q_8\}$ and $Q_{acc} = \{q_9, q_{10}, q_{11}\}$. The starting state of K_2 is q_{12}. The transition matrixes $V_{\looparrowright}$, V_a, V_b and $V_{\looparrowleft}$ are defined as follows:

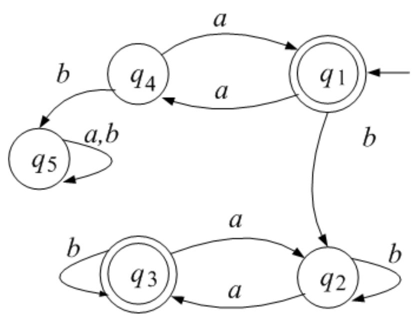

Figure 3.9: Minimal Automaton of L_2 "even"

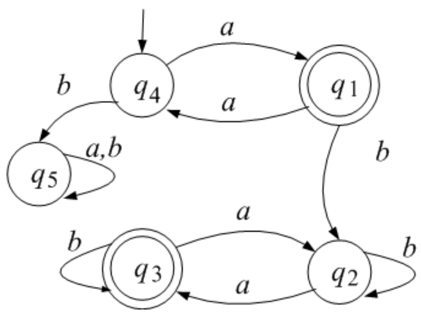

Figure 3.10: Minimal Automaton of L_3 "odd"

$$V_{\looparrowright} = \begin{pmatrix}
\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} \\
0 & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

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

$$V_b = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$V_{\hookleftarrow} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

1. After reading the left endmarker $\hookrightarrow$ K_2 with probability $\frac{2}{3}$ is in the state q_1 and with probability $\frac{1}{3}$ is in the state q_2. G_2 is in the starting state q_1.

2. After reading even number of letters a K_2 with probability $\frac{2}{3}$ is in the state q_1 and with probability $\frac{1}{3}$ is in the state q_2.

3. After reading odd number of letters a K_2 with probability $\frac{2}{3}$ is in the state q_4 and with probability $\frac{1}{3}$ is in the state q_3.

4. If after reading an odd number of the letter a K_2 receives the letter b or right endmarker then it rejects input with probability at least $\frac{2}{3}$ (from the state q_4 by reading b or right endmarker K_2 goes to rejecting state)

5. If after reading even number of letters a K_2 receives right endmarker then it accepts the input with probability $\frac{2}{3}$

6. If after reading even number of letters a K_2 receives letter b then with probability $\frac{1}{3}$ K_2 passes to accepting state, with probability $\frac{1}{3}$ K_2 passes to rejecting state, and probability $\frac{1}{3}$ K_2 passes to the non-final state q_2

7. By reading the letter a automaton K_2 passes from q_2 to q_3 or back. By reading the letter b automaton K_2 passes from q_2 to q_2 and from q_3 to q_3, so receiving right endmarker in the state q_3 the input is accepted with total probability $\frac{2}{3}$ and receiving right endmarker in the state q_2 the input is rejected with total probability $\frac{2}{3}$.

   This shows that K_2 accepts the language L_2 with probability $\frac{2}{3}$. Similarly we construct K_3 that accepts L_3 with probability $\frac{2}{3}$.

   Thus we have shown that there are two languages L_2 and L_3 which are recognizable by DH-PRA with probability $\frac{2}{3}$, but the union of them $L_1 = L_2 \cup L_3$ is not recognizable by DH-PRA.

$\square$

3.5 Classification of Reversible Automata

In this section we summarize the models of 1-way reversible automata and their computational power in a table, providing references to papers where they have been considered.

	C-Automata	DH-Automata
Deterministic Automata	Permutation Automata [HS 66, T 68] (C-DRA)	Reversible Finite Automata [AF 98] (DH-DRA)
Quantum Automata with Pure States	Measure-Once Quantum Finite Automata [MC 97] (C-QFA-P)	Measure-Many Quantum Finite Automata [KW 97] (DH-QFA-P)
Probabilistic Automata	Probabilistic Reversible C-Automata (C-PRA)	Probabilistic Reversible DH-Automata (DH-PRA)
Quantum Finite Automata with Mixed States	"Latvian" QFA [ABGKMT 06] (C-QFA-M)	Enhanced Quantum Finite Automata [N 99] (DH-QFA-M)

Language class problems have been solved for C automata and DH-DRA, for the remaining types of DH automata they are still open. Every type of DH-automata may simulate the corresponding type of C-automata.

The following relation among language classes also presents interest, question marks denoting conjectures:

$$\text{C-DRA} = \text{C-QFA-P} \subset \text{C-PRA} = \text{C-QFA-M}$$
$$\text{DH-DRA} \subset \text{DH-QFA-P} \overset{?}{\subset} \text{DH-PRA} \overset{?}{\subset} \text{DH-QFA-M}$$

Generally, language classes recognized by C-automata are closed under boolean operations, while DH-automata are not (open for the DH-QFA-M).

Most recent result [ABGKMT 06] proved that classes of languages recognizable by C-PRA and C-QFA-M are equal and coincide with the all regular languages but languages which minimal deterministic automaton contains forbidden constructions considered in the thesis.

3.6 Weak reversibility

3.6.1 Reversibility and weak reversibility

Notion of reversibility implies ability to get input from the result. That is quite straightforward in deterministic case but no so obvious in probabilistic case, where the definition mimics quantum one, based on sum of probabilities to access particular configuration from the other to be one. So natural way of thinking about reversibility is the ability for automaton to work into the opposite direction.

Definition 3.34. *An automaton of some type is called **weakly reversible** if the reverse of its transition function*[1] *corresponds to the transition function of a valid automaton of the same type.*

In case of transition function for 1-way automata $\delta : Q \times \Gamma \times Q \longrightarrow \mathbb{R}_{[0,1]}$ reverse function $\delta' : Q \times \Gamma \times Q \longrightarrow \mathbb{R}_{[0,1]}$ is such that for any states q and q' and letters σ, $\delta'(q, \sigma, q') = \delta(q', \sigma, q)$. In case of one-way automata it is easy to check that this definition is equivalent to the one in section 3.3.

Note: in case of deterministic automaton where $\delta : Q \times \Gamma \times Q \longrightarrow \{0, 1\}$ this property means that automaton is still deterministic not nondeterministic.

We give an example that illustrates that in case of 1.5-way automata these definitions are different.

[1]by reverse of automata function we understand the function defined such that all transitions between states change their direction

3.6.2 1.5-way Probabilistic Reversible Automata

Definition 3.35. *1.5-way probabilistic* ***weakly*** *reversible C-automaton $A = (Q, \Sigma, q_0, Q_F, \delta)$ is specified by Q, Σ, q_0, Q_F defined as in 1-way C-PRA definition 3.2, and a transition function*

$$\delta : Q \times \Gamma \times Q \times D \longrightarrow \mathbb{R}_{[0,1]},$$

where Γ defined as in 1-way C-PRA definition and $D = \{0, 1\}$ denotes whether automaton stays on the same position or moves one letter ahead on the input tape. Furthermore, transition function satisfies the following requirements:

$$\forall (q_1, \sigma_1) \in Q \times \Gamma \sum_{q \in Q, d \in D} \delta(q_1, \sigma_1, d, q) = 1 \tag{3.18}$$

$$\forall (q_1, \sigma_1) \in Q \times \Gamma \sum_{q \in Q, d \in D} \delta(q, \sigma_1, d, q_1) = 1 \tag{3.19}$$

Definition 3.36. *1.5-way probabilistic reversible C-automaton $A = (Q, \Sigma, q_0, Q_F, \delta)$ is specified by Q, Σ, q_0, Q_F defined as in 1-way C-PRA definition 3.2, and a transition function*

$$\delta : Q \times \Gamma \times Q \times D \longrightarrow \mathbb{R}_{[0,1]},$$

where Γ defined as in 1-way C-PRA definition and $D = \{0, 1\}$ denotes whether automaton stays on the same position or moves one letter ahead on the input tape. Furthermore, transition function satisfies the following requirements:

$$\forall (q_1, \sigma_1) \in Q \times \Gamma \sum_{q \in Q, d \in D} \delta(q_1, \sigma_1, q, d) = 1 \tag{3.20}$$

$$\forall (q_1, \sigma_1, \sigma_2) \in Q \times \Gamma^2 \sum_{q \in Q} \delta(q, \sigma_1, q_1, 0) + \sum_{q \in Q, \sigma \in \Gamma} \delta(q, \sigma_2, q_1, 1) = 1 \tag{3.21}$$

Theorem 3.37. *Language (a,b)*a is recognizable by 1.5-way weak C-PRA.*

Proof. The $Q = \{q_0, q_1\}$, $Q_F = \{q_1\}$, δ is defined as follows

$\delta(q_0, a, 0, q_0) = \frac{1}{2}$ $\delta(q_0, a, 1, q_1) = \frac{1}{2}$ $\delta(q_1, a, 0, q_0) = \frac{1}{2}$ $\delta(q_1, a, 1, q_1) = \frac{1}{2}$

$\delta(q_0, b, 1, q_0) = \frac{1}{2}$ $\delta(q_0, b, 0, q_1) = \frac{1}{2}$ $\delta(q_1, b, 1, q_0) = \frac{1}{2}$ $\delta(q_1, b, 0, q_1) = \frac{1}{2}$

$\delta(q_0, \hookleftarrow, 1, q_0) = 1$ $\delta(q_1, \hookleftarrow, 1, q_1) = 1$

It easy to check that such automaton moves ahead according to the transition of the following deterministic automaton

$$\begin{array}{cc} \delta(q_0, a, 1, q_1) = 1 & \delta(q_1, a, 1, q_1) = 1 \\ \delta(q_0, b, 1, q_0) = 1 & \delta(q_1, b, 1, q_0) = 1 \\ \delta(q_0, \hookleftarrow, 1, q_0) = 1 & \delta(q_1, \hookleftarrow, 1, q_1) = 1 \end{array}$$

So the probability of wrong answer is 0. The probability to be at the m position of the input tape after n steps of calculation for $m \leq n$ is C_n^m. Therefore it is necessary no more then $O(n * \log(p))$ steps to reach the end of the word of length n (and since obtain correct answer) with probability $1 - \frac{1}{p}$

□

Still this result is of limited nature.

Chapter 4

Quantum one way 1 counter automata

4.1 Definition of Q1CA

4.1.1 Classical 1CA

Definition 4.1. *A one-counter deterministic finite automaton (D1CA) A is specified by the finite (input) alphabet* Σ*, the finite set of states* Q*, the initial state* q_0*, the sets* $Q_a \subset Q$ *and* $Q_r \subset Q$ *of accepting and rejecting states, respectively, with* $Q_a \cap Q_r = \emptyset$*, and the transition function* $\delta\colon Q \times \Sigma \times S \to Q \times \{\leftarrow\downarrow\rightarrow\}$*, where* $S = \{0, 1\}$*.*

Additionally to the keeping the state the automaton is in and position on the input tape, there is a counter holding an arbitrary integer. The counter is set to zero at the beginning of computation. The *transition function* determines how the state and the counter value are updated as input letters are read from the tape. $\leftarrow, \downarrow, \rightarrow$, mean, respectively, decrease by one, retain the same and increase by one the value of the counter. The value of transition function is determined by the current letter of the input, state automaton is in and whether the counter is zero on non-zero not the exactly value of the counter. Thus S is defined to be 0 if and only if the value of the counter is equal to 0, otherwise equal to 1. The computation of the input word is done letter-wise until the last letter in the word is reached. If the automaton is then in an accepting state, the word is considered accepted, otherwise, the word is rejected.

Thus such automata can be viewed as special case of pushdown automata where stack alphabet contains only one symbol and a special marker of the bottom (automata allowing negative integers can be simulated with ones not allowing). The D1CA recognizes proper subset of context free languages. It has been

Definition 4.2. *A probabilistic finite one-counter automaton (P1CA) A is specified by the finite (input) alphabet Σ, the finite set of states Q, the initial state q_0, the sets $Q_a \subset Q$ and $Q_r \subset Q$ of accepting and rejecting states, respectively, with $Q_a \cap Q_r = \emptyset$, and the transition function $\delta\colon Q \times \Sigma \times S \times Q \times \{\leftarrow\downarrow\rightarrow\} \to R^+$, where $S = \{0, 1\}$ and δ satisfies the following condition:*

$$\sum_{q',d} \delta(q, \sigma, s, q', d) = 1 \quad \textit{for each} \;\; q, q' \in Q, \sigma \in \Gamma, s \in \{0, 1\}, d \in \{\leftarrow, \downarrow, \rightarrow\}.$$

Example 4.3. Non-context free P1CA can recognize the language L_2 $0^n10^n10^n$ with probability $1-1/n$, for each $n \in N, n \geq 2$ [Fr 78]. The basic idea of the automaton is that the probabilistic decision is made during the first step and one of the following n paths is chosen with equal probability. Each path is a deterministic automaton that accepts the word if it is in the form $0^i10^j10^k$ and an equation in the form $a*i+b*j = (a+b)*k$, where $a, b \in N$ is satisfied. We can choose such a, b for each path that the equation can be satisfied at most in one path for any word, which is in form $0^i10^j10^k$ and does not belong to L_2. Thus if the word belongs to L_2 than the automaton accepts it with probability 1. If the word is not like $0^i10^j10^k$ than it is rejected with probability 1. If the word does not belong to L_2 but is like $0^i10^j10^k$ then it is rejected with probability at least $1 - 1/n$.

4.1.2 General model of Q1CA

Definition 4.4. *A quantum finite one-counter automaton (Q1CA) A is specified by the finite (input) alphabet Σ, the finite set of states Q, the initial state q_0, the sets $Q_a \subset Q$ and $Q_r \subset Q$ of accepting and rejecting states, respectively, with $Q_a \cap Q_r = \emptyset$, and the transition function $\delta\colon Q \times \Gamma \times S \times Q \times \{\leftarrow \downarrow \rightarrow\} \rightarrow C$, where $\Gamma = \Sigma \cup \{⤙, ⤚\}$ is the tape alphabet of A and symbols $⤙, ⤚$ are left and right endmarkers not in Σ, $S = \{0,1\}$, and δ satisfies the following conditions (of well-formedness) for each $q_1, q_2, q' \in Q, \sigma \in \Gamma, s \in \{0,1\}, d \in \{\leftarrow, \downarrow, \rightarrow\}$:*

1. *Local probability and orthogonality condition*

$$\sum_{q',d} \delta^*(q_1, \sigma, s_1, q', d)\delta(q_2, \sigma, s_2, q', d) = \begin{cases} 1, & \textit{if } q_1 = q_2 \\ 0, & \textit{if } q_1 \neq q_2 \end{cases} \tag{4.1}$$

2. *Separability condition I*

$$\begin{aligned} &\sum_{q',d} \delta^*(q_1, \sigma, s_1, q', \rightarrow)\delta(q_2, \sigma, s_2, q', \downarrow) + \\ +&\sum_{q',d} \delta^*(q_1, \sigma, s_1, q', \downarrow)\delta(q_2, \sigma, s_2, q', \leftarrow) = 0 \end{aligned} \tag{4.2}$$

3. *Separability condition II*

$$\sum_{q',d} \delta^*(q_1, \sigma, s_1, q', \rightarrow)\delta(q_2, \sigma, s_2, q', \longleftarrow) = 0, \tag{4.3}$$

where * denotes complex conjunctive.

Formally $A = (\Sigma, Q, q_0, Q_a, Q_r, \delta)$.

For an integer n let C_n be the set of all possible configurations of A for inputs of length n. The definition determines that at the n-th step automata reads n-th symbol of w_x, and before the n-th step the counter can contain value from $-(n-1)$ up to $n-1$. So the configuration of A for each specific input x at each step can be uniquely determined by a pair $(q,k), q \in Q$ and $k \in [0, n-1]$, where q is the state of the automata and k is value of the counter.

A computation of A on an input x of length n corresponds to a unitary evolution in the underlying Hilbert space $H_{A,n} = l_2(C_n)$. For each $c \in C_n, |c\rangle$ denotes the basis vector in $l_2(C_n)$, we will use also $|q,k\rangle$. Each state in $H_{A,n}$ will therefore have a form $\sum_{c \in C_n} \alpha_c |c\rangle$, where $\sum_{c \in C_n} |\alpha_c|^2 = 1$. The automaton A induces for any input $x \in \Sigma^n$ a linear operator U_x^δ that is defined as

$$U_x^\delta \left|q, k\right\rangle = \sum_{q',d} \delta(q, w_{xi}, sign(k), q', d) \left|q', k + \mu(d)\right\rangle \tag{4.4}$$

for a configuration $(q, k) \in C_n$, where w_{xi} denotes i-th symbol of $w_x = \looparrowright x \looparrowleft$, $\text{sign}(k) = 0$ if $k = 0$ and 1 otherwise, $\mu(d) = -1(0)[1]$ if $d = \leftarrow (\downarrow)[\rightarrow]$. By linearity U_x^δ is extended to map any superposition of basis states.

This definition corresponds to usual DH model of quantum automata. If we are not restricted with n instead of set of all possible configurations C to be used instead ofC_n.

4.1.3 Unitarity of Q1CA

Now we will prove that the evolution of a Q1CA A, satisfying conditions 1 - 3, is unitary. We consider the ultimate case when no restriction on input length. As underlying Hilbert space is not finite dimensional we need to prove both $U_x^{\delta^*} U_x^\delta = I$ and $U_x^\delta U_x^{\delta^*} = I$. But we should note that transformation matrix is of special form

- finite number of nonzero element in each row and column as we can get by one step to the configuration with counter value different at most by one
- there is finite number of different rows and columns as δ is independent on exact value of the counter, all the other rows and columns can be received from them by offset of them.

For such matrix we have seen that $U_x^\delta U_x^{\delta^*} = I$ follows from $U_x^{\delta^*} U_x^\delta = I$ and norm of the vectors equal to 1 (see Preliminaries Lemma 2.7).

We will prove that using the approach similar to used in [BV 97] for Quantum Turing Machine.

Lemma 4.5. *Let U be a transition matrix of Q1CA and $U^*U = I$ then norm of any row vector of U is equal to 1.*

Proof. For arbitrary n columns of U let B be a m × n matrix that contains all the rows having nonzero elements when intersecting with selected n columns. Assume for any arbitrary small ϵ we can select such n and matrix B that 2 conditions hold:

- $m/n \leq 1 + \epsilon$

- there is a constant μ that rows of particular type from U are in B at least m/μ times

Then the sum of squares of B elements is equal to the n if counting by column norms. Let for some row in B the norm to be $1-\delta$ and the row is in B at least m/μ times, then we can get a contradiction by calculating the same sum by m rows, that can not be greater then:
$m*(1-1/\mu)+m/\mu(1-1/\delta) \leq m-m(\delta/\mu) \leq n(1+\epsilon)(1-\delta/\mu)$. If $\epsilon < \delta/(\mu-\delta)$ then $n(1+\epsilon)(1-\delta/\mu) < n$ that will give us required contradiction.

To construct such a matrix B let us select value of the counter from c to c+k-1. Number of configurations S that correspond to these values is $n = card(Q) * k$. Columns in U that correspond to the configurations S hold non zero values on the rows that correspond to the configurations from S and those having value of the counter c-1 and c+k additionally. The number of later is $2 * card(Q)$. Thus $m = (k+2)card(Q)$ and we can select k large enough to get for arbitrary small ϵ $m/n = 1 + 2/k \leq 1 + \epsilon$. In such B the rows that correspond to the counter values from c+1 till c+k-2 hold nonzero elements only in columns from B , these rows make up $1 - 2/k$ of all rows, thus μ can be selected. So we get required construction of B for all the rows excluding for those corresponding to c and c+k-1 value of the counter. But for those we can select another c so that these rows would be internal. $\square$

Lemma 4.6. *For any input string x $U_x^{\delta^*} U_x^\delta = I$ iff the conditions (1) to (3) of Definition 4.4 are satisfied.*

Proof. $U_x^{\delta^*} U_x^\delta = I$ can be rewritten as:

1. $\left\| U_x^\delta \left|q,k\right\rangle \right\| = \left\langle U_x^\delta \left|q,k\right\rangle \middle| U_x^\delta \left|q,k\right\rangle \right\rangle = 1$, for all configurations (q,k);
2. $U_x^\delta \left|q_1,k_1\right\rangle \perp U_x^\delta \left|q_2,k_2\right\rangle$ for all different configurations (q_1,k_1) and (q_2,k_2) the last can be written as
   (a) $\left\langle U_x^\delta \left|q_1,k_1\right\rangle \middle| U_x^\delta \left|q_2,k_2\right\rangle \right\rangle = 0$
   (b) $\left\langle U_x^\delta \left|q_2,k_2\right\rangle \middle| U_x^\delta \left|q_1,k_1\right\rangle \right\rangle = 0$,

   where $q, q_1, q_2 \in Q$ and $k, k_1, k_2 \in [0, |x|+1]$.

The $\left\langle U_x^\delta \left|q_1,k_1\right\rangle \middle| U_x^\delta \left|q_2,k_2\right\rangle \right\rangle$ can be rewritten as

$$\left\langle U_x^\delta \left|q_1,k_1\right\rangle \middle| U_x^\delta \left|q_2,k_2\right\rangle \right\rangle =$$

$$\sum_{q',d_1,d_2} \delta^*(q_1, w_{xi}, sign(k_1), q', d_1)\delta(q_2, w_{xi}, sign(k_2), q', d_2)$$

where $k_1 + d_1 = k_2 + d_2$. Each member of the sum corresponds to the product of the amplitudes of $U_x^{\delta *}|q_1, k_1\rangle$ and $U_x^{\delta}|q_2, k_2\rangle$ mapping to the same configuration $|q', k_1 + d_1\rangle = |q', k_2 + d_2\rangle$.

If conditions 1. to 3. of Definition 4.4 are satisfied, then

1. $\left\|U_x^{\delta}|q, k\rangle\right\| = \sum_{q',d} \delta^*(q, w_{xi}, sign(k), q', d)\delta(q, w_{xi}, sign(k), q', d) = 1$, when 1. is true for $q_1 = q_2$

2. We observe separately the following cases $(k_1 \leq k_2)$ (for $(k_2 < k_1)$ it can be shown similar):

   2.1. $k_1 = k_2$ $(q_1 \neq q_2)$

   $$(2a) = \sum_{q',d} \delta^*(q_1, w_{xi}, sign(k), q', d)\delta(q_2, w_{xi}, sign(k), q', d) = 0$$

   when 1. is true for $q_1 \neq q_2$.

   2.2. $k_2 - k_1 > 2$ $(2a) = (2b) = 0$, because there is no such $|q', k'\rangle$ for which there is non zero amplitude in both $U_x^{\delta *}|q_1, k_1\rangle$ and $U_x^{\delta}|q_2, k_2\rangle$ in this case, because the value of the counter can change at most by 1 at each step.

   2.3. $k_2 - k_1 = 2$

   $$(2a) = \sum_{q'} \delta^*(q_1, w_{xi}, sign(k_1), q', \rightarrow)\delta(q_2, w_{xi}, sign(k_2), q', \leftarrow) = 0$$

   if 3. is true.

   2.4. $k_2 - k_1 = 1$

   $$(2a) = \sum_{q'} \delta^*(q_1, w_{xi}, sign(k_1), q', \downarrow)\delta(q_2, w_{xi}, sign(k_2), q', \leftarrow) + \sum_{q'} \delta^*(q_1, w_{xi}, sign(k_1), q', \rightarrow)\delta(q_2, w_{xi}, sign(k_2), q', \downarrow) = 0$$

   when 2. is true.

These equalities stand in case if $U_x^{\delta^*} U_x^{\delta} = I$ thus to prove lemma into the opposite direction we would consider the same cases. □

So we can formulate the theorem

Theorem 4.7. *For any input string x the mapping U_x^{δ} is unitary if and only if the conditions (1) to (3) of Definition 4.4 are satisfied.*

Proof. Follows from Lemmas 4.6 and 4.5 and 2.7 □

4.2 Models of Q1CA

4.2.1 Simple Q1CA

Conditions 1–3 are not constructive, so given an automaton we can test them, but they do not allow us to specify the automaton in constructive way easily. So like the definition of simple 2-way QFA, see [KW 97], we can define *simple Q1CA*.

Definition 4.8. *A Q1CA is simple, if for each $\sigma \in \Gamma, s \in \{0,1\}$ there is a linear unitary operator $V_{\sigma,s}$ on the inner product space $l_2(Q)$ and a function $D\colon Q,\Gamma \to \{\leftarrow,\downarrow,\rightarrow\}$ such that for each $q \in Q, \sigma \in \Gamma, s \in \{0,1\}$*

$$\delta(q,\sigma,s,q',d) = \begin{cases} \langle q'|V_{\sigma,s}|q\rangle & \text{if } D(q',\sigma)=d \\ 0 & \text{else} \end{cases}$$

where $\langle q'|V_{\sigma,s}|q\rangle$ denotes the coefficient of $|q'\rangle$ in $V_{\sigma,s}\,|q\rangle$.

In other words that means that transition function is defined by

- (a) Separate unitary matrixes for each letter of the working alphabet and also for zero and non-zero value of the counter that determine change of the state.
- (b) Function that determine the change of the value of the counter by the state automaton moves in and letter read.

Theorem 4.9. *A simple Q1CA satisfies the well-formedness conditions 1–3. if and only if*

$$\sum_{q'} \langle q'|V_{\sigma,s}|q_1\rangle^* \langle q'|V_{\sigma,s}|q_2\rangle = \begin{cases} 1, \text{if } q_1 = q_2 \\ 0, \text{if, } q_1 \neq q_2 \end{cases}$$

for each $\sigma \in \Gamma, s \in \{0,1\}$. That holds if and only if every operator is unitary.

Proof. We can simply rewrite well-formedness conditions:

$$\sum_{q',d} \delta^*(q_1,\sigma,s,q',d)\delta(q_2,\sigma,s,q',d) =$$

$$\sum_{q'} \delta^*(q,\sigma,s,q',D(q',d))\delta(q,\sigma,s,q',D(q',d)) + 0 =$$

$$\sum_{q'} \langle q'|V_{\sigma,s}|q_1\rangle^* \langle q'|V_{\sigma,s}|q_2\rangle$$

Conditions 2. and 3. are satisfied because $D(q', d)$ can be equal only to one of the $\leftarrow, \downarrow, \rightarrow$ for all δ in the sum. But each member of the sums 2., 3. has two multipliers with different d ($\leftarrow, \downarrow$ and $\downarrow, \rightarrow$ in 2. and $\leftarrow, \rightarrow$ in 3.). So each member of the sum always has at least one of the multipliers equal to 0. Thus the whole sum is 0 too. We can use these considerations also to prove the theorem in the opposite direction. □

4.2.2 Models of acceptance by Q1CA

In this section we define formally the acceptance and rejection models of Q1CA. An observable used in Q1CA is defined the way for any DH automata. The difference is that in a one counter automata case acceptance can be defined in several ways:

1. acceptance both by state and zero value of the counter
2. acceptance by zero value of the counter
3. acceptance by state

For each input word x with $n = |x|$ and a Q1CA $A = (\Sigma, Q, q_0, Q_a, Q_r, \delta)$ let

$$\begin{aligned}
C_n^a &= \left\{ (q,k) \mid (q,k) \in C_n, \begin{array}{ll} 1. & q \in Q_a, k = 0 \\ 2. & k = 0 \\ 3. & q \in Q_a \end{array} \right\} \\
C_n^r &= \left\{ (q,k) \mid (q,k) \in C_n, \begin{array}{ll} 1. & q \in Q_r \\ 2. & q \in Q_r, k \neq 0 \\ 3. & q \in Q_r \end{array} \right\} \\
C_n^- &= C_n - C_n^a - C_n^r.
\end{aligned}$$

Let E_a, E_r and E_- be the subspaces of $l_2(C_n)$ spanned by C_n^a, C_n^r and C_n^- respectively.

The "computational observable" Ω corresponds to the orthogonal decomposition $l_2(C_n) = E_a \oplus E_r \oplus E_-$. The outcome of any observation will be either "accept" (E_a) or "reject" (E_r) or "non-terminating" (E_-).

The language recognition by A is now defined as follows: for an $x \in \Sigma^*$ as the input is used $w_x = \hookrightarrow x \hookleftarrow$, computation starts in the state $|q_0, 0\rangle$ and counter is set to 0. For each letter from w_x operator U_x^δ is applied to current state and the resulting state is observed using the computational observable Ω defined above. After it the state collapses into E_a, or E_r or E_-. If "non-terminating" state is observed than computation continues with next letter. The probability of the acceptance, rejection and non-terminating at each step is equal to the square of amplitude of new state for the corresponding subspace. Computation stops either after halting state is observed or word is proceeded.

In the thesis we use language recognition by state and counter value as more widely spread and corresponding to classical definition.

4.3 Non-Context-free language recognition by Q1CA

4.3.1 Languages.

Consider the alphabet $\Sigma = \{0, 1\}$.

The language L_1. $L_1 = \{\ 0^i10^j10^k \parallel$ (i=k or j=k) and $\neg$(i=j)) $\}$.

The language L_2. $L_2 = \{\ 0^i10^j10^k \parallel$ exactly 2 of i,j,k are equal$\}$.

Both languages are non context-free , that can be checked by using Pumping Lemma, see [Gu 89]

4.3.2 Results.

Theorem 4.10. *The language L_1 can be recognized by quantum one-counter one-way automata with probability $\frac{5}{8}$.*

Proof. Let $V_{\hookleftarrow,0}|q_0\rangle = \frac{\sqrt{5}}{4}\,|q_{0,i=k}\rangle + \frac{\sqrt{5}}{4}\,|q_{0,j=k}\rangle + \frac{\sqrt{6}}{4}\,\left|q_{0,k=(i+j)/2}\right\rangle$, where q_0 is initial state, $q_{0,j=k}$, $q_{0,i=k}$ and $q_{0,k=(i+j)/2}$ non-terminating states. Transitions for 0 and 1 can be defined in such way, that it would be reversible and deterministic and the following conditions are satisfied if starting state of such deterministic automaton is $q_{0,j=k}$, $q_{0,i=k}$ or $q_{0,k=(i+j)/2}$ respectively:
1) If the word is of form $0^i10^j10^k$ that no rejection or acceptance occur during the computation and the $q_{0,j=k}$ leads to the state $q_{j=k}$ and counter equal to the j-k, the $q_{0,i=k}$ to $q_{i=k}$ and counter i-k, the $q_{0,k=(i+j)/2}$ to $q_{k=(i+j)/2}$ and counter k - (i+j)/2.
2) If the word is not like $0^i10^j10^k$ than the word is rejected in each path, or is in some other state then $q_{i=k}$, $q_{j=k}$ and $q_{k=(i+j)/2}$ and thus will be rejected on end marker $\hookleftarrow$.
Such transitions for example for $q_{0,j=k}$, will be as follows: 4 states $q_{0,j=k}$, $q_{1,j=k}$, $q_{2,j=k}$, $q_{3,j=k}$. Transitions for 0 are defined as each state remains the same and counter value is increased if resulting state is $q_{1,j=k}$, and decreased if $q_{2,j=k}$; transitions for 1 defined as states should shift to the next index in respect with module 4 so $q_{n,j=k}$ should shift to $q_{n+1 mod 4,j=k}$ and counter value

remains the same. $q_{3,j=k}$ should be rejecting state, $q_{2,j=k} = q_{j=k}$.
We define V for $\hookleftarrow$ as
$V_{\hookleftarrow,0} \left|q_{i=k}\right\rangle = \sqrt{\frac{3}{5}} \left|q_{a1}\right\rangle + \frac{1}{\sqrt{5}} \left|q_a\right\rangle + \frac{1}{\sqrt{5}} \left|q_r\right\rangle$;
$V_{\hookleftarrow,0} \left|q_{j=k}\right\rangle = \sqrt{\frac{3}{5}} \left|q_{a2}\right\rangle - \frac{1}{\sqrt{5}} \left|q_a\right\rangle + \frac{1}{\sqrt{5}} \left|q_r\right\rangle$;
$V_{\hookleftarrow,0} \left|q_{k=(i+j)/2}\right\rangle = \left|q_{r2}\right\rangle$;
$V_{\hookleftarrow,1} \left|q_{k=(i+j)/2}\right\rangle = \left|q_{a2}\right\rangle$
where q_a, q_{a1}, q_{a2} are accepting states and q_r , q_{r2} are rejecting states. All the other transition should be defined to map any of non-halting states to the rejecting states (it can be done by adding some more rejecting states to retain unitarity see [K 99]).
Let us consider how the computation goes with such automata. Before reading $\hookleftarrow$ word is rejected if it is not of kind $0^i10^j10^k$, and is in the superposition $\left|q'\right\rangle = \frac{\sqrt{5}}{4} \left|q_{i=k}, i-k\right\rangle + \frac{\sqrt{5}}{4} \left|q_{j=k}, j-k\right\rangle + \frac{\sqrt{6}}{4} \left|q_{k=(i+j)/2}, k-(i+j)/2\right\rangle$ otherwise.
We should consider following cases then:

- 1. If i=j=k then the state after reading $\hookleftarrow$ becomes
$\frac{\sqrt{5}}{4}\left(\sqrt{\frac{3}{5}} \left|q_{a1}, 0\right\rangle + \frac{1}{\sqrt{5}} \left|q_a, 0\right\rangle + \frac{1}{\sqrt{5}} \left|q_r, 0\right\rangle\right) +$
$+ \frac{\sqrt{5}}{4}\left(\sqrt{\frac{3}{5}} \left|q_{a2}, 0\right\rangle + \frac{1}{\sqrt{5}} \left|q_a, 0\right\rangle + \frac{1}{\sqrt{5}} \left|q_r, 0\right\rangle\right) + \frac{\sqrt{6}}{4} \left|q_r, 0\right\rangle =$
$\frac{\sqrt{3}}{4} \left|q_{a1}, 0\right\rangle + \frac{\sqrt{3}}{4} \left|q_{a2}, 0\right\rangle + \frac{1}{2} \left|q_r, 0\right\rangle + \frac{\sqrt{6}}{4} \left|q_r, 0\right\rangle$.
Thus the total probability of rejection, by summing squares of amplitudes from rejecting states is $\frac{5}{8}$.

- 2. (i=k)and¬(i=j) then the state is
$\frac{\sqrt{5}}{4}\left(\sqrt{\frac{3}{5}} \left|q_{a1}, 0\right\rangle + \frac{1}{\sqrt{5}} \left|q_a, 0\right\rangle + \frac{1}{\sqrt{5}} \left|q_r, 0\right\rangle\right) +$
$+ \frac{\sqrt{5}}{4} \left|q_r, j-k\right\rangle + \frac{\sqrt{6}}{4} \left|q_a, (j-k)/2\right\rangle$.
So the word is accepted with $p = \frac{3}{16} + \frac{1}{16} + \frac{6}{16} = \frac{5}{8}$.

- 3. (j=k)and(¬(i=j)). The same as shown in the previous item.

- 4. If all i, j, k are different, then we should distinguish 2 subcases
  - a) if not (i+j)/2=k then word is rejected with probability 1 then word is rejected with probability 1 due to construction of automata - all other transitions from non-halting states to rejecting states.
  - b) if (i+j)/2 =k in this case word is rejected with probability $1-\left(\frac{\sqrt{6}}{4}\right)^2 = \frac{5}{8}$.

So the automaton recognizes L_1 with probability $\frac{5}{8}$. Note that this probability is higher then $\frac{3}{5}$ found in [YKTI 00].

□

Theorem 4.11. *The language* L_2 *can be recognized by quantum one-counter one-way automata with probability* 0.58.

Proof. Let $V_{\looparrowleft,0}|q_0\rangle = \frac{1}{\sqrt{5}}|q_{0,i=k}\rangle + \frac{1}{\sqrt{5}}|q_{0,j=k}\rangle + \frac{1}{\sqrt{5}}|q_{0,i=j}\rangle +$

$+\frac{\sqrt{2}}{\sqrt{5}}\left|q_{0,k=(i+j)/2}\right\rangle$, where q_0 is initial state, $q_{0,j=k}$, $q_{0,i=k}$, $q_{0,i=j}$, and $q_{0,k=(i+j)/2}$ non-terminating states. Transitions for 0 and 1 can be defined in such way, that it would be reversible and deterministic and the following conditions are satisfied if starting state of such deterministic automaton is $q_{0,j=k}$, $q_{0,i=k}$, $q_{0,i=j}$ or $q_{0,k=(i+j)/2}$ respectively:
1) If the word is in form $0^i10^j10^k$ that no rejection or acceptance occur during the computation and the $q_{0,j=k}$ leads to the state $q_{j=k}$ and counter equal to the j-k, the $q_{0,i=k}$ to $q_{i=k}$ and counter i-k, $q_{0,i=j}$ to $q_{i=j}$ and counter i-j, the $q_{0,k=(i+j)/2}$ to $q_{k=(i+j)/2}$ and counter k - (i+j)/2.
2) If the word is not like $0^i10^j10^k$ than the word is rejected in each path, or is in some other state then $q_{i=j}$, $q_{i=k}$, $q_{j=k}$ and $q_{k=(i+j)/2}$ and thus will be rejected on end marker $\looparrowleft$.
(Such transitions can be easily defined like shown in proof of Theorem 1).

We define V for $\looparrowleft$ as
$V_{\looparrowleft,0}\,|q_{i=k}\rangle = \frac{1}{\sqrt{10}}\left(\sqrt{7}\,|q_{a,i=k}\rangle + |q_r\rangle + |q_{a1}\rangle + |q_{a2}\rangle\right)$;
$V_{\looparrowleft,0}\,|q_{j=k}\rangle =$

$\frac{1}{\sqrt{10}}\left(\sqrt{7}\,|q_{a,j=k}\rangle + |q_r\rangle + \left(-\frac{1}{2} + \frac{\sqrt{3}}{2}i\right)|q_{a1}\rangle + \left(-\frac{1}{2} - \frac{\sqrt{3}}{2}i\right)|q_{a2}\rangle\right)$;
$V_{\looparrowleft,0}\,|q_{i=j}\rangle =$

$\frac{1}{\sqrt{10}}\left(\sqrt{7}\,|q_{a,i=j}\rangle + |q_r\rangle + \left(-\frac{1}{2} - \frac{\sqrt{3}}{2}i\right)|q_{a1}\rangle + \left(-\frac{1}{2} + \frac{\sqrt{3}}{2}i\right)|q_{a2}\rangle\right)$;
$V_{\looparrowleft,0}\,\left|q_{k=(i+j)/2}\right\rangle = |q_{r2}\rangle$; $V_{\looparrowleft,1}\,\left|q_{k=(i+j)/2}\right\rangle = |q_{a1}\rangle$,
where $q_{a,i=k}$, $q_{a,i=j}$, $q_{a,j=k}$, q_{a1}, q_{a2} are accepting states and q_r , q_{r2} are rejecting states. All the other transition should be defined to map any of non-halting states to the rejecting states (it can be done by adding some more rejecting states to retain unitarity, see [K 99])

/TODO Describe it in some appendix!

Let us consider how the computation goes with such automata. Before reading $\looparrowleft$ word is rejected if it is not of kind $0^i10^j10^k$, and is in the superposition
$|q'\rangle = \frac{1}{\sqrt{5}}|q_{i=k}, i-k\rangle + \frac{1}{\sqrt{5}}|q_{j=k}, j-k\rangle + \frac{1}{\sqrt{5}}|q_{i=j}, i-j\rangle +$
$+\frac{\sqrt{2}}{\sqrt{5}}\left|q_{k=(i+j)/2}, k-(i+j)/2\right\rangle$.

We should consider then the following cases

- 1. If i=j=k then the state after reading $\hookleftarrow$ due to the sum up of the amplitudes with same states becomes
$\frac{1}{\sqrt{50}}\left(\sqrt{7}\,|q_{a,j=k},0\rangle+\sqrt{7}\,|q_{a,i=k},0\rangle+\sqrt{7}\,|q_{a,i=j},0\rangle+3\,|q_r,0\rangle\right)+$
$\sqrt{\frac{2}{5}}\,|q_{r2},0\rangle$. The total probability of rejection is $\frac{9}{50}+\frac{2}{5}$=0.58.
- 2. (i=k)and¬(i=j) then the state is
$\frac{1}{\sqrt{50}}\left(\sqrt{7}\,|q_{a,j=k},0\rangle+|q_{r,},0\rangle+|q_{a1},0\rangle+3\,|q_{a2},0\rangle\right)+$
$+\ \frac{1}{\sqrt{5}}\,|q_{r,i=j},i-j\rangle\ +\ \frac{1}{\sqrt{5}}\,|q_{r,j=k},j-k\rangle\ +\ \sqrt{\frac{2}{5}}\,|q_a,(j-k)/2\rangle$.
So the word is accepted with p=$\frac{7}{50}$+$\frac{1}{50}$+$\frac{1}{50}$+$\frac{2}{5}$=0.58.
- 3. (j=k)and¬(i=j) The same as shown in the previous item.
- 4. (i=j)and¬(i=k) The same as shown in the previous item.
- 5. If all i, j, k are different, then we should distinguish 2 subcases:
  - a) if not (i+j)/2=k then word is rejected with probability 1 due to construction of automata - all other transitions from non-halting states to rejecting states.
  - b) if (i+j)/2 =k in this case word is rejected with probability 1-$\left(\frac{\sqrt{2}}{5}\right)^2$=0.6.

So the automaton recognizes L_2 with probability 0.58. Note that this probability is higher then $\frac{4}{7}$ found in [YKTI 00].

$\square$

4.4 Q1CA versus P1CA

In this section we prove the existence of languages that are recognized by Q1CA but not with P1CA.

4.4.1 Languages.

Let Σ be a finite alphabet. For $S \subseteq \Sigma$ define the 'projection' map $\pi_S : \Sigma^* \rightarrow S^*$ which acts on words over Σ by forgetting all letters not in S. When S is given explicitly as $\{\sigma_1,...,\sigma_n\}$, we write $\pi_{\sigma_1,...,\sigma_n}$. Note, in particular, that the length $|\pi_\sigma(x)|$ counts the occurrence of a letter $\sigma \in \Sigma$ in a word $x \in \Sigma^*$.

The language L_1. Consider the alphabet $\Sigma = \{0, 1, 2, \flat, 3, 4, 5, \sharp\}$ with two special symbols $\flat$ and $\sharp$. Suppose we formally decompose the set of all words in each of the subalphabets $\Sigma^\flat = \{0, 1, 2, \flat\}$ and $\Sigma^\sharp = \{3, 4, 5, \sharp\}$,

$$\Sigma^{\flat *} = \Lambda_1^\flat \cup \Lambda_2^\flat \cup \Lambda_3^\flat \quad \textit{and} \quad \Sigma^{\sharp *} = \Lambda_1^\sharp \cup \Lambda_2^\sharp \cup \Lambda_3^\sharp, \tag{4.5}$$

and put, for short,

$$L_i^\flat = \pi_{\Sigma^\flat}^{-1}(\Lambda_i^\flat) \quad \textit{and} \quad L_i^\sharp = \pi_{\Sigma^\sharp}^{-1}(\Lambda_i^\sharp), \quad i = 1, 2, 3. \tag{4.6}$$

Note that Σ^* is then decomposed into eight components $L_i^\flat \cap L_j^\sharp$, $i, j = 1, 2, 3$. Define formally the language L_1 as the union of two of these:

$$L_1 = (L_1^\flat \cap L_2^\sharp) \cup (L_2^\flat \cap L_1^\sharp). \tag{4.7}$$

The decompositions (4.5) are now set as follows. Let $\Lambda^\flat$ denote the set of all words in $\{0, 1, 2, \flat\}^*$ of the form $x\flat y$, where $x \in \{0, 1\}^*$, $y \in \{2\}^*$, and put

$$\Lambda_1^\flat = \{x \in \Lambda^\flat : |\pi_0(x)| = |\pi_1(x)|\}$$

and,

$$\Lambda_2^\flat = \{x \in \Lambda^\flat : |\pi_0(x)| = |\pi_1(x)| + |\pi_2(y)| \quad \textit{and} \quad |\pi_2(y)| > 0\}.$$

Analogously, for the alphabet $\{3, 4, 5, \sharp\}$,

$$\Lambda_1^\sharp = \{x \in \Lambda^\sharp : |\pi_3(x)| = |\pi_4(x)|\}$$

and

$$\Lambda_2^\sharp = \{x \in \Lambda^\sharp : |\pi_3(x)| = |\pi_4(x)| + |\pi_5(y)| \quad \textit{and} \quad |\pi_5(y)| > 0\}.$$

The language L_2. Additional symbols α, β_1, β_2 are added to the alphabet of L_1. A word is in L_2 if it is of the form

$$x_1 \ \alpha y_1 \ x_2 \ \alpha y_2 \ x_3 \ \alpha y_3 \ldots \ \alpha y_{n-1} \ x_n \ \alpha \tag{4.8}$$

with $x_1, x_2, \ldots, x_n$ in L_1 and $y_i{=}\beta_1$ iff x_i is in $(L_1^\flat \cap L_2^\sharp)$, $y_i{=}\beta_2$ iff x_i is in $(L_2^\flat \cap L_1^\sharp)$.

4.4.2 Results

Theorem 4.12. *The language L_1 cannot be recognized by deterministic one-counter one-way automata, but it can be recognized with bounded error by a quantum one-counter one-way automaton.*

Proof. To prove the first claim, assume to the contrary that a deterministic one-counter one-way finite automaton recognizing L_1 does exist and has k states. Consider the words $x_{ij} = 0^i 3^j 1^i 4^{j-1} \flat\sharp 5$, where $j \leq i \leq n$, and n is some large integer. Clearly, all $x_{ij} \in L_1$. When the first part $0^i 3^j$ of the word x_{ij} has been read, the value of the counter is at most $i+j \leq 2n$, so the automaton can at this stage distinguish at most $2nk$ of the words x_{ij} which are $\frac{1}{2}n(n+1)$ in total. Thus, if n is large enough, two different words, $x_{i_1 j_1}$ and $x_{i_2 j_2}$, say, would, at this stage of computation, share the same state and counter value. But then, clearly, the automaton would also accept the words $0^{i_1} 3^{j_1} 1^{i_2} 4^{j_2-1} \flat\sharp 5$ and $0^{i_2} 3^{j_2} 1^{i_1} 4^{j_1-1} \flat\sharp 5$ neither of which is in the language L_1.

We now briefly describe a quantum one-counter one-way automaton which, as we subsequently show, recognizes L_1 with bounded error. In addition to an initial state, the automaton will have sixteen non-terminating states q_{ijk}, q'_{ij}, q''_{ij}, $i, j, k = 1, 2$, four accepting states $a_1, \ldots, a_4$, and eight rejecting states $r_1, \ldots, r_8$. As customary, we interpret invertible transformations of the set of basis states of the automaton as unitary operators in its quantum configuration space.

When the initial marker $\looparrowright$ comes in, the states q_{ij1} get amplitudes $(-1)^{i+j}\frac{1}{2}$, while all the remaining states get amplitude zero.When any of the the symbols $0, 1, 3$, or 4 arrives, the state remains unchanged; the counter is changed only in the following cases: the symbols 0 and 3 *increase* the counter for the states q_{1jk} and q_{2jk}, respectively, while the symbols 1 and 4 *decrease* the counter for the states q_{1jk} and q_{2jk}, respectively.

The special symbol $\flat$ is ignored if read in any of the states q_{2jk} or q''_{ij}; if read in state q_{1jk} and the counter is empty, the state q'_{jk} follows, while the state q'_{jk^*}[1] follows if the counter is non-empty; if read in state q'_{ij}, the rejecting states $r_1, \ldots, r_4$ follow.

The special symbol $\sharp$ is ignored if read in any of the states q_{1jk} or q'_{ij}; if read in state q_{2jk} and the counter is empty, the state q''_{jk} follows, while the state q''_{jk^*} follows if the counter is non-empty; if read in state q''_{ij}, the rejecting states $r_5, \ldots, r_8$ follow.

The symbol 2 is ignored if read in any of the states q_{2jk}, q''_{ij}, or q'_{1j}; if read in state q_{1jk} the rejecting states $r_1, \ldots, r_4$ follow; if read in state q'_{2j}, the

[1] If e is an element of a two-element set (here $\{1,2\}$), we write e^* to denote the other of the two elements.

state remains unchanged while the counter decreases.

The symbol 5 is ignored if read in any of the states q_{1jk}, q'_{ij}, or q''_{1j}; if read in state q_{2jk} the rejecting states $r_1, \ldots, r_4$ follow; if read in state q''_{2j}, the state remains unchanged while the counter decreases.

When the end marker $\hookleftarrow$ arrives, if the value of the counter is zero, a unitary transformation is applied consistently with the following transition table:

	a_1	a_2	r_1	r_2	a_3	a_4	r_3	r_4
q'_{11}	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
q'_{12}	-$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	-$\frac{1}{2}$	0	0	0	0
q'_{21}	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
q'_{22}	0	0	0	0	$\frac{1}{2}$	-$\frac{1}{2}$	$\frac{1}{2}$	-$\frac{1}{2}$
q''_{11}	$\frac{1}{2}$	-$\frac{1}{2}$	$\frac{1}{2}$	-$\frac{1}{2}$	0	0	0	0
q''_{12}	-$\frac{1}{2}$	-$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
q''_{21}	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	-$\frac{1}{2}$	-$\frac{1}{2}$
q''_{22}	0	0	0	0	$\frac{1}{2}$	-$\frac{1}{2}$	-$\frac{1}{2}$	$\frac{1}{2}$

To complete the description of the automaton we should extend the transitions given above to describe the whole automaton as follows:

- Each non terminal state in the case that there is no transformation given above for some letter and zero and non zero counter value separately, should be mapped to some rejecting state. It can be easily accomplished, so that do not violate unitarity by adding some more rejecting states.

- The transitions for each letter and zero or non zero counter value separately for cases that are not described should be specified arbitrary, so that ensure the unitarity of the matrix of corresponding transformation for this letter and counter value.

It remains to verify that the automaton just described indeed recognizes the language L_1 with bounded error. It is not hard to compute the nonzero amplitudes for the automaton's states when a word $x \in \Sigma^*$ has been processed, just before the end marker $\hookleftarrow$ arrives. We do this for each of the eight cases as $x \in L_i^\flat \cap L_j^\sharp$, $i, j = 1, 2, 3$. We look first at the cases $i, j = 1, 2$. The value of the counter is 0 in all these cases. Note that in the case when x is an empty word, it has the same distribution of amplitudes as in $x \in L_1^\flat \cap L_1^\sharp$, so we will consider these two cases together.

	q'_{11}	q'_{12}	q'_{21}	q'_{22}	q''_{11}	q''_{12}	q''_{21}	q''_{22}
$x \in L_1^\flat \cap L_1^\sharp$	$\frac{1}{2}$	0	-$\frac{1}{2}$	0	-$\frac{1}{2}$	0	$\frac{1}{2}$	0
$x \in L_1^\flat \cap L_2^\sharp$	$\frac{1}{2}$	0	-$\frac{1}{2}$	0	0	-$\frac{1}{2}$	0	$\frac{1}{2}$
$x \in L_2^\flat \cap L_1^\sharp$	0	$\frac{1}{2}$	0	-$\frac{1}{2}$	-$\frac{1}{2}$	0	$\frac{1}{2}$	0
$x \in L_2^\flat \cap L_2^\sharp$	0	$\frac{1}{2}$	0	-$\frac{1}{2}$	0	-$\frac{1}{2}$	0	$\frac{1}{2}$

Straightforward calculation then gives the following non-zero amplitudes after the end marker $\hookleftarrow$ has been processed:.

	a_1	a_2	r_1	r_2	a_3	a_4	r_3	r_4
$x \in L_1^\flat \cap L_1^\sharp$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	-$\frac{1}{2}$	-$\frac{1}{2}$
$x \in L_1^\flat \cap L_2^\sharp$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	-$\frac{1}{2}$	-$\frac{1}{2}$	0
$x \in L_2^\flat \cap L_1^\sharp$	-$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	-$\frac{1}{2}$	0
$x \in L_2^\flat \cap L_2^\sharp$	0	$\frac{1}{2}$	0	-$\frac{1}{2}$	0	0	-$\frac{1}{2}$	$\frac{1}{2}$

Hence, after the measurement, the accepting probability for $x \in L_1 = (L_1^\flat \cap L_2^\sharp) \cup (L_2^\flat \cap L_1^\sharp)$ is equal to $\frac{3}{4}$, while for $x \in L_1^\flat \cap L_1^\sharp$ or $x \in L_2^\flat \cap L_2^\sharp$ it is equal to $\frac{1}{4}$; the corresponding rejecting probabilities are complementary.

It remains to check the cases when i or j in $L_i^\flat \cap L_j^\sharp$ is equal to three. The amplitudes for the non-terminal states and zero value of the counter just before the end marker $\hookleftarrow$ arrives are then as follows:

	q'_{11}	q'_{12}	q'_{21}	q'_{22}	q''_{11}	q''_{12}	q''_{21}	q''_{22}
$x \in L_1^\flat \cap L_3^\sharp$	$\frac{1}{2}$	0	-$\frac{1}{2}$	0	0	0	0	0
$x \in L_2^\flat \cap L_3^\sharp$	0	$\frac{1}{2}$	0	-$\frac{1}{2}$	0	0	0	0
$x \in L_3^\flat \cap L_1^\sharp$	0	0	0	0	-$\frac{1}{2}$	0	$\frac{1}{2}$	0
$x \in L_3^\flat \cap L_2^\sharp$	0	0	0	0	0	-$\frac{1}{2}$	0	$\frac{1}{2}$

In the remaining case $x \in L_3^\flat \cap L_3^\sharp$ all non-terminal amplitudes are zero.

Hence, after the end marker has been processed, we have the following terminal amplitudes:

	a_1	a_2	r_1	r_2	a_3	a_4	r_3	r_4
$x \in L_1^\flat \cap L_3^\sharp$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$
$x \in L_2^\flat \cap L_3^\sharp$	-$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$
$x \in L_3^\flat \cap L_1^\sharp$	-$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$
$x \in L_3^\flat \cap L_2^\sharp$	$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$

and in the case $x \in L_3^\flat \cap L_3^\sharp$ all non-terminal amplitudes are still zero. The probability to accept a word x in any of these four cases is equal to $\frac{1}{4}$, while in the case $x \in L_3^\flat \cap L_3^\sharp$ it is zero; the rejecting probabilities are complementary.

Hence, summing up, the automaton accepts all words in the language L_1 with probability $\frac{3}{4}$, and rejects all words not in L_1 with probability at least $\frac{3}{4}$.

□

Theorem 4.13. *The language L_2 cannot be recognized with bounded error by probabilistic one-counter one-way finite automata, but it can be recognized with probability $\frac{3}{4}$ by a quantum one-counter one-way finite automaton.*

Proof. For the first statement, we first note that the language L_1 cannot be recognized with probability one by a probabilistic one-way one-counter finite automaton. Indeed, assuming the contrary and simulating the probabilistic automaton by a deterministic one (our automaton reads one input symbol at a time, so we may take the first available choice at any time), would bring us into contradiction with the first part of Theorem 1. The impossibility to recognize L_2 by a probabilistic automaton with a bounded error now follows, since the subwords $x_i \in L_1$ of a word x in L_2 can be taken in arbitrarily large numbers, and every x_i is processed with a positive error.

For the second part of the theorem, we extend the construction of the quantum automaton described in the proof of Theorem 1. Our extended automaton is to read the symbols $\alpha, \beta_1, \beta_2,$. We need four other non-terminating states q_i for it.

The transformation for the α is described as follows.

$$\begin{array}{c|cccccccc} & q_1 & q_2 & r_1 & r_2 & r_3 & q_3 & q_4 & r_4 \\ \hline q'_{11} & \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ q'_{12} & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ q'_{21} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ q'_{22} & 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} \\ q''_{11} & 0 & \frac{1}{\sqrt{2}} & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ q''_{12} & -\frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ q''_{21} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} \\ q''_{22} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{2} \end{array}$$

The transformation for states $q1, q_2, q3, q_4$ and zero value of the counter for both letters β_1 and β_2, can be written with one table, the only difference is the resulting states.

(β_1)	q_{111}	q_{121}	q_{211}	q_{221}	r_1	r_2	r_3	r_4
(β_2)	r_1	r_2	r_3	r_4	q_{111}	q_{121}	q_{211}	q_{221}
q_1	$\frac{1}{\sqrt{2}}$	0	0	$\frac{1}{\sqrt{2}}$	0	0	0	0
q_2	0	0	0	0	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0
q_3	0	0	0	0	$\frac{1}{\sqrt{2}}$	0	0	$\frac{1}{\sqrt{2}}$
q_4	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	0	0	0	0

Finally we describe the transformation for the final marker $\hookleftarrow$ for states $q1$, q_2, q_3, q_4 and zero value of the counter.

	a_1	a_2	a_3	r_1
q_1	1	0	0	0
q_2	0	1	0	0
q_3	0	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
q_4	0	0	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$

To complete the description of the automaton we should extend described transformations for each letter and zero and non zero value of the counter separately the same way as in the proof of Theorem 1.

It remains to verify that the automaton just described indeed recognizes the language L_2 with probability $\frac{3}{4}$.

While processing x_1 the automaton acts the same as described for L_1 so when first α comes the distribution of amplitudes is exactly the same as described in proof of L_1 before $\hookleftarrow$. So after applying transformation for α the automaton gets the following amplitudes for states with zero value of the counter.

	q_1	q_2	r_1	r_2	r_3	q_3	q_4	r_4	p reject
$x_1 \in L_1^\flat \cap L_1^\sharp$	$\frac{1}{2\sqrt{2}}$	$-\frac{1}{2\sqrt{2}}$	0	$\frac{1}{2}$	0	$\frac{1}{2\sqrt{2}}$	$-\frac{1}{2\sqrt{2}}$	$-\frac{1}{2}$	$\frac{1}{2}$
$x_1 \in L_1^\flat \cap L_2^\sharp$	$\frac{1}{\sqrt{2}}$	0	0	0	0	0	$-\frac{1}{\sqrt{2}}$	0	0
$x_1 \in L_2^\flat \cap L_1^\sharp$	0	$-\frac{1}{\sqrt{2}}$	0	0	0	$\frac{1}{\sqrt{2}}$	0	0	0
$x_1 \in L_2^\flat \cap L_2^\sharp$	$\frac{1}{2\sqrt{2}}$	$-\frac{1}{2\sqrt{2}}$	0	$-\frac{1}{2}$	0	$\frac{1}{2\sqrt{2}}$	$-\frac{1}{2\sqrt{2}}$	$\frac{1}{2}$	$\frac{1}{2}$
$x_1 \in L_1^\flat \cap L_3^\sharp$	$\frac{1}{2\sqrt{2}}$	0	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	0	$-\frac{1}{2\sqrt{2}}$	$-\frac{1}{4}$	$\frac{3}{4}$
$x_1 \in L_2^\flat \cap L_3^\sharp$	0	$\frac{1}{2\sqrt{2}}$	$\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	0	$\frac{1}{2\sqrt{2}}$	$\frac{1}{4}$	$\frac{3}{4}$
$x_1 \in L_3^\flat \cap L_1^\sharp$	0	$-\frac{1}{2\sqrt{2}}$	$-\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2\sqrt{2}}$	0	$-\frac{1}{4}$	$\frac{3}{4}$
$x_1 \in L_3^\flat \cap L_2^\sharp$	$\frac{1}{2\sqrt{2}}$	0	$-\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$	0	$-\frac{1}{2\sqrt{2}}$	$\frac{1}{4}$	$\frac{3}{4}$
$x_1 \in L_3^\flat \cap L_3^\sharp$	0	0	0	0	0	0	0	0	1

The last column shows the total probability of rejection after processing $x_1\alpha$.

The resulting amplitudes for states with the value of the counter 0 are the following

	q_1	q_2	q_3	q_4
$x_1 \in L_1^\flat \cap L_1^\sharp$	$\frac{1}{2\sqrt{2}}$	-$\frac{1}{2\sqrt{2}}$	$\frac{1}{2\sqrt{2}}$	-$\frac{1}{2\sqrt{2}}$
$x_1 \in L_1^\flat \cap L_2^\sharp$	$\frac{1}{\sqrt{2}}$	0	0	-$\frac{1}{\sqrt{2}}$
$x_1 \in L_2^\flat \cap L_1^\sharp$	0	-$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0
$x_1 \in L_2^\flat \cap L_2^\sharp$	$\frac{1}{2\sqrt{2}}$	-$\frac{1}{2\sqrt{2}}$	$\frac{1}{2\sqrt{2}}$	-$\frac{1}{2\sqrt{2}}$
$x_1 \in L_1^\flat \cap L_3^\sharp$	$\frac{1}{2\sqrt{2}}$	0	0	-$\frac{1}{2\sqrt{2}}$
$x_1 \in L_2^\flat \cap L_3^\sharp$	0	$\frac{1}{2\sqrt{2}}$	0	$\frac{1}{2\sqrt{2}}$
$x_1 \in L_3^\flat \cap L_1^\sharp$	0	-$\frac{1}{2\sqrt{2}}$	$\frac{1}{2\sqrt{2}}$	0
$x_1 \in L_3^\flat \cap L_2^\sharp$	$\frac{1}{2\sqrt{2}}$	0	0	-$\frac{1}{2\sqrt{2}}$
$x_1 \in L_3^\flat \cap L_3^\sharp$	0	0	0	0

So we see that only in the cases $x_1 \in L_1^\flat \cap L_2^\sharp$ or $x_1 \in L_2^\flat \cap L_1^\sharp$ the probability of rejection is 0. When i or j in $L_i^\flat \cap L_j^\sharp$ is equal to three the probability of rejection is at least $\frac{3}{4}$, so these cases are not considered further. When $x_1 \in L_1^\flat \cap L_1^\sharp$ or $x_1 \in L_2^\flat \cap L_2^\sharp$ the probability of rejection is $\frac{1}{2}$.

If β_1 or $\beta 2$ come after α then the amplitudes become

(β_1) (β_2)	q_{111} r_1	q_{121} r_2	q_{211} r_3	q_{221} r_4	r_1 q_{111}	r_2 q_{121}	r_3 q_{211}	r_4 q_{221}
$x_1 \in L_1^\flat \cap L_1^\sharp$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$
$x_1 \in L_1^\flat \cap L_2^\sharp$	$\frac{1}{2}$	-$\frac{1}{2}$	-$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
$x_1 \in L_2^\flat \cap L_1^\sharp$	0	0	0	0	$\frac{1}{2}$	-$\frac{1}{2}$	-$\frac{1}{2}$	$\frac{1}{2}$
$x_1 \in L_2^\flat \cap L_2^\sharp$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	-$\frac{1}{4}$	-$\frac{1}{4}$	$\frac{1}{4}$

So if x_1 is in $(L_1^\flat \cap L_2^\sharp)$ and β_1 is read or x_1 is in $(L_2^\flat \cap L_1^\sharp)$ and β_2 is read then we get the same amplitude distribution as after initial marker $\looparrowright$.

If x_1 is in $(L_1^\flat \cap L_2^\sharp)$ and β_2 is read or x_1 is in $(L_2^\flat \cap L_1^\sharp)$ and β_1 is read then the word is rejected with probability 1.

If x_1 is in $(L_1^\flat \cap L_1^\sharp)$ or x_1 is in $(L_2^\flat \cap L_2^\sharp)$ then $\frac{1}{2}$ of the remaining amplitudes is in rejecting states. So thus after α the probability of rejection for such a word is already $\frac{1}{2}$, the probability becomes $\frac{3}{4}$

We should also check cases when β_1 or β_2 occur in another position than after α. Due to the construction of automaton the word is rejected immediately.

So we have seen that after processing $x_1\alpha y_1$ the automaton is in the same quantum state as after reading initial marker $\looparrowright$ in the cases x_1 is in $(L_1^\flat \cap L_2^\sharp)$ and y_1=β_1 or x_1 is in $(L_2^\flat \cap L_1^\sharp)$ and y_1= β_2 and the word is rejected with probability at least $\frac{3}{4}$ otherwise.

So the computation on $x_i\alpha y_i$ will be the same as for $x_1\alpha y_1$ if the previous part of the word corresponds to the conditions of the language and the word will be rejected with probability at least $\frac{3}{4}$ otherwise.

Finally we need to show the processing of the final marker. Note that it should come after α otherwise the word is rejected. We consider the case when the final marker comes after $x_1\alpha$.

The resulting amplitudes are

	a_1	a_2	a_3	r_1
$x_1 \in L_1^\flat \cap L_1^\sharp$	$\frac{1}{2\sqrt{2}}$	-$\frac{1}{2\sqrt{2}}$	0	-$\frac{1}{2}$
$x_1 \in L_1^\flat \cap L_2^\sharp$	$\frac{1}{\sqrt{2}}$	0	-$\frac{1}{2}$	$\frac{1}{2}$
$x_1 \in L_2^\flat \cap L_1^\sharp$	0	-$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	$\frac{1}{2}$
$x_1 \in L_1^\flat \cap L_1^\sharp$	$\frac{1}{2\sqrt{2}}$	-$\frac{1}{2\sqrt{2}}$	0	-$\frac{1}{2}$

So the word is accepted with probability $\frac{3}{4}$ in the $x_1 \in L_1^\flat \cap L_2^\sharp$ and $x_1 \in L_2^\flat \cap L_1^\sharp$ cases. The word is rejected with probability $\frac{1}{2}$ already after processing α in the other two cases described above and thus the total probability of rejection is $\frac{3}{4}$.

□

4.5 Restrictions of Q1CA

There is proven that Q1CA can not recognize all regular languages. That follows from general results on the quantum automata by Nayak [N 99]. As noted in [YKTI 00] Q1CA can not recognize language $(a|b)^*\,a$ as dimension of the quantum system that coincide with configurations reachable on the input word of length n is card(Q)*n, but according to Nayak's results it should be exponential.

Bibliography

[AKN 97] D. Aharonov, A. Kitaev and N. Nisan. Quantum circuits with mixed states. *30th Annual ACM Symposium on Theory of Computation*, pp. 20-30, 1998.

[AI 99] M. Amano, K. Iwama. Undecidability on Quantum Finite Automata. *31st STOC*, pp. 368-375, 1999.

[ABGKMT 06] A. Ambainis, M. Beaudry, M. Golovkins, A. Kikusts, M. Mercer, D. Thrien. Algebraic results on quantum automata. Theory of Computing Systems, 39, pp. 165-188, 2006. Earlier version in STACS'04
`http://www.math.uwaterloo.ca/%7Eambainis/ps/lqfa.ps`

[ABFK 99] A. Ambainis, R. Bonner, R. Freivalds, A. Ķikusts. Probabilities to Accept Languages by Quantum Finite Automata. *COCOON 1999, Lecture Notes in Computer Science*, Vol. 1627, pp. 174-183, 1999.
`http://arxiv.org/abs/quant-ph/9904066`

[AF 98] A. Ambainis, R. Freivalds. 1-Way Quantum Finite Automata: Strengths, Weaknesses and Generalizations. *Proc. 39th FOCS*, pp. 332-341, 1998.
`http://arxiv.org/abs/quant-ph/9802062`

[AKV 00] A. Ambainis, A. Ķikusts, M. Valdats. On the Class of Languages Recognizable by 1-Way Quantum Finite Automata. *STACS 2001, Lecture Notes in Computer Science*, Vol. 2010, pp. 75-86, 2001.
`http://arxiv.org/abs/quant-ph/0009004`

[ANTV 98] A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani. Dense Quantum Coding and a Lower Bound for 1-Way Quantum Automata. *Proc. 31st STOC*, pp. 376-383, 1999.
`http://arxiv.org/abs/quant-ph/9804043`

[AW 99] A. Ambainis, J. Watrous. Two-Way Finite Automata with Quantum and Classical States.
`http://arxiv.org/abs/cs.CC/9911009`

[BV 97] E. Bernstein, U. Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, Vol. 26(5), pp. 1411-1473, 1997.

[BBBV 97] C. Bennett, E. Bernstein, G. Brassard, U. Vazirani. Strengths and Weaknesses of Quantum Computing *SIAM Journal on Computing*, Vol. 26 (5), pp. 1510 - 1523, 1997.

[BMP 03] Alberto Bertoni, Carlo Mereghetti, and Beatrice Palano. Quantum Computing: 1-Way Quantum Automata, *DLT 2003, Lecture Notes in Computer Science*, Vol. 2710, pp. 120, 2003.

[BFK 01] R. Bonner, R. Freivalds, M. Kravtsev. Quantum versus Probabilistic One-Way Finite Automata with Counter. *SOFSEM 2001, Lecture Notes in Computer Science*, Vol. 2234, pp. 181-190, 2001.

[BP 99] A. Brodsky, N. Pippenger. Characterizations of 1-Way Quantum Finite Automata.
`http://arxiv.org/abs/quant-ph/9903014`

[De 85] D. Deutsch. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society (London)*, Vol. A-400, pp. 97-117, 1985.

[DSa 96] C. Dürr, M. Santha. A Decision Procedure for Unitary Linear Quantum Cellular Automata. *Proc. 37th FOCS*, pp. 38-45, 1996.
`http://arxiv.org/abs/quant-ph/9604007`

[Du 01] A. Dubrovsky. Space-Efficient 1.5-Way Quantum Turing Machine. *FCT 2001, Lecture Notes in Computer Science*, Vol. 2138, pp. 380-383, 2001.

[Dz 03] I. Dzelme. Kvantu automāti ar jauktajiem stāvokļiem. *Latvijas Universitāte*, bakalaura darbs.

[DSt 89] C. Dwork, L. Stockmeyer. On the Power of 2-Way Probabilistic Finite State Automata. *Proc. 30th FOCS*, pp. 480-485, 1989.

[Fr 78] R. Freivalds. Recognition of Languages with High Probability on Different Classes of Automata. *Doklady Akad. Nauk SSSR*, Vol. 239(1), pp. 60-62, 1978. (Russian)
Also *Soviet Math. Doklady*, Vol. 19(2), pp. 295-298, 1978.

[Fr 81] R. Freivalds. Probabilistic Two-Way Machines. *MFCS 1981, Lecture Notes in Computer Science*, Vol. 118, pp. 33-45, 1981.

[FGK 04] R. Freivalds, M.Golovkins, A. Ķikusts. On the Properties of Probabilistic Reversible Automata. *SOFSEM 2004, Student Forum*, pp. 78 - 87, 2004.

[Go 00] M. Golovkins. Quantum Pushdown Automata. *SOFSEM 2000, Lecture Notes in Computer Science*, Vol. 1963, pp. 336-346, 2000.
`http://arxiv.org/abs/quant-ph/0102054`

[Go 02] M Golovkins. Quantum Automata and Quantum Computing. Doctoral Thesis, University of Latvia, 2002.

[GK 02] M. Golovkins, M. Kravtsev. Probabilistic Reversible Automata and Quantum Automata. *COCOON 2002, Lecture Notes in Computer Science*, Vol. 2387, pp. 574-583, 2002.
`http://arxiv.org/abs/cs.CC/0209018`

[GKK 06] M.Golovkins, M. Kravcevs, V. Kravcevs. On the Class of Languages Recognizable by Probabilistic Reversible Decide-and-Halt Automata. *Submitted to SWAT 2006 - 10th Scandinavian Workshop on Algorithm Theory*, 12 pages, 2006.

[GKK 05] M.Golovkins, M. Kravcevs, V. Kravcevs. On the Class of Languages Recognizable by Probabilistic Reversible Decide-and-Halt Automata. Extended Abstract. *5th int. ERATO Conference on Quantum Information Systems. Proceedings, ERATO project*, pp. 131-132, 2005.

[GS 97] C. M. Grinstead and J. L. Snell. Introduction to Probability. *American Mathematical Society*, 1997.
`http://www.dartmouth.edu/~chance/teaching_aids/articles.html`

[Gr 99] J. Gruska. Quantum Computing, McGraw Hill, 439 p., 1999.

[G 97] L.Grover Quantum Mechanics Helps in Searching for a Needle in a Haystack Phys. Rev. Lett. 79, pp. 325 - 328, 1997.
`http://arxiv.org/abs/quant-ph/9706033`

[Gu 89] E. Gurari. An Introduction to the Theory of Computation. *Computer Science Press*, 1989.

[HS 66] J. Hartmanis and R.E.Stearns. Algebraic Structure Theory of Sequential Machines. *Prentice Hall*, 1966.

[KS 76] J. G. Kemeny and J. L. Snell. Finite Markov Chains. *Springer Verlag*, 1976.

[KW 97] A. Kondacs, J. Watrous. On The Power of Quantum Finite State Automata. *Proc. 38th FOCS*, pp. 66-75, 1997.

[K 99] M. Kravtsev. Quantum Finite One-Counter Automata. *SOFSEM 1999, Lecture Notes in Computer Science*, Vol. 1725, pp. 431-440, 1999. `http://arxiv.org/abs/quant-ph/9905092`

[K 04] M. Kravtsev. Better Probabilities for One-Counter Quantum Automata. *6th International Baltic Conference on Data Bases and Information Systems. Proceedings, University of Latvia*, pp. 128-135, 2004.

[MO 79] A. Marshall, I. Olkin. Inequalities: Theory of Majorization and Its Applications. *Academic Press*, 1979.

[MC 97] C. Moore, J. P. Crutchfield. Quantum automata and quantum grammars. *Theoretical Computer Science*, Vol. 237(1-2), pp. 275-306, 2000.
`http://arxiv.org/abs/quant-ph/9707031`

[Mi 67] M. Minsky. Computation: Finite and Infinite Machines, *Prentice-Hall, Inc., N.J.*, 1967.

[Mo 96] . Morita. Universality of a Reversible Two-Counter Machine *Theoretical Computer Science*, Vol. 168, pp. 303-320, 1996.

[N 99] A. Nayak. Optimal Lower Bounds for Quantum Automata and Random Access Codes. *Proc. 40th FOCS*, pp. 369-377, 1999.
`http://arxiv.org/abs/quant-ph/9904093`

[NC 00] M. Nielsen,I. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 675 p.,2000.

[R 63] M. O. Rabin. Probabilistic Automata. *Information and Control*, Vol. 6(3), pp. 230-245, 1963.

[Si 94] D. Simon, On the power of quantum computation,*35th Annual IEEE Symposium on Foundations of Computer Science*, pp. 116-123, 1994.

[Sh 94] P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. *Proc. 35th FOCS*, pp. 124-134, 1994.
`http://arxiv.org/abs/quant-ph/9508027`

[T 68] G. Thierrin. Permutation Automata. *Mathematical Systems Theory*, Vol. 2(1), pp. 83-90, 1968.

[Y 93] A. Yao, Quantum Circuit Complexity, *Proc. 34th IEEE Symp. on Foundations of Computer Science*, pp. 352-361, 1993.

[YKI 02] T. Yamasaki, H. Kobayashi, H. Imai. Quantum versus Deterministic Counter Automata. *COCOON 2002, Lecture Notes in Computer Science*, Vol. 2387, pp. 584-594, 2002.

[YKTI 00] T. Yamasaki, H. Kobayashi, Y. Tokunaga, H. Imai. One-Way Probabilistic Reversible and Quantum One-Counter Automata. *COCOON 2000, Lecture Notes in Computer Science*, Vol. 1858, pp. 436-446, 2000.